
Efficient Amortised Bayesian Inference for Hierarchical and Nonlinear Dynamical Systems

Geoffrey Roeder^{1,2} Paul K Grant¹ Andrew Phillips¹ Neil Dalchau¹ Edward Meeds¹

Abstract

We introduce a flexible, scalable Bayesian inference framework for nonlinear dynamical systems characterised by distinct and hierarchical variability at the individual, group, and population levels. Our model class is a generalisation of nonlinear mixed-effects (NLME) dynamical systems, the statistical workhorse for many experimental sciences. We cast parameter inference as stochastic optimisation of an end-to-end differentiable, block-conditional variational autoencoder. We specify the dynamics of the data-generating process as an ordinary differential equation (ODE) such that both the ODE and its solver are fully differentiable. This model class is highly flexible: the ODE right-hand sides can be a mixture of user-prescribed or “white-box” sub-components and neural network or “black-box” sub-components. Using stochastic optimisation, our amortised inference algorithm could seamlessly scale up to massive data collection pipelines (common in labs with robotic automation). Finally, our framework supports interpretability with respect to the underlying dynamics, as well as predictive generalization to unseen combinations of group components (also called “zero-shot” learning). We empirically validate our method by predicting the dynamic behaviour of bacteria that were genetically engineered to function as biosensors.

1. Introduction

Dynamical systems have been the central focus in many areas of statistical machine learning, whether as state-space

¹Microsoft Research, Cambridge, United Kingdom ²Princeton University, Princeton, United States of America. Correspondence to: Geoffrey Roeder <roeder@princeton.edu>, Ted Meeds <edmeeds@microsoft.com>, Neil Dalchau <ndalchau@microsoft.com>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

models (Archer et al., 2015; Karl et al., 2016), (non)linear dynamical systems and their extensions (Becker-Ehmck et al., 2018; Johnson et al., 2016; Linderman et al., 2016), or neuroscience applications (Wu et al., 2017; Gao et al., 2016). Recent work has even reinterpreted general recursive deep neural networks as discretisations of continuous-time dynamical systems (Chen et al., 2018).

Dynamical systems learned from experimental data are widespread in the physical sciences, including fluid dynamics, thermodynamics, and electromagnetism. They also play a particularly important role in advancing our understanding of biology, typically studied as Ordinary Differential Equations (ODEs). Seminal contributions of ODE models in biology include the mechanisms of the cell cycle (Tyson et al., 2001; Ferrell et al., 2011), circadian rhythms (Leloup & Goldbeter, 2003), and nerve action potentials (Hodgkin & Huxley, 1952).

More recently, ODE models have been playing an increasingly important role in engineering biological systems by capturing mechanistic processes in the underlying chemical reaction dynamics that identify potential targets for molecular intervention and by predicting the possible outcomes of these interventions (Elowitz & Leibler, 2000; Potvin-Trottier et al., 2016; Gardner et al., 2000). The ability to precisely engineer biology could enable substantial breakthroughs in medicine and crop yields, and provide sustainable alternatives to environmentally unsustainable processes and products (Khalil & Collins, 2010; Ruder et al., 2011; Cumbers & Schmieder, 2017). However, identifying the parameters of ODE models is still limited by the efficiency and scalability of inference methods (Calderhead & Girolami, 2011). Markov Chain Monte Carlo (MCMC) is considered state-of-the-art for learning ODE parameters from experimental data (Woods et al., 2016; Moore et al., 2018), but requires expensive numerical integration at each time step.

This problem is exacerbated by model selection, where competing explanatory mechanisms must be evaluated for predictive accuracy and generalisation (Calderhead & Girolami, 2009). In the era of increasingly automated laboratories, scientists require new statistical models and computational methods that can scale adequately with the increasing availability of experimental data.

To address these computational and modeling challenges, we introduce **VI-HDS**: Variational Inference over the parameters of **Hierarchical Dynamical Systems** that have a defined generative process and likelihood. Specifically, we cast inference over the parameters of a dynamical system with hierarchical or multi-level structure as optimisation of a conditional variational autoencoder (Sohn et al., 2015; Kingma & Welling, 2013; Rezende et al., 2014).

We interpret this framework as a generalisation of “nonlinear mixed-effects models” (NMEMs) (Davidian & Giltinan, 2003)—a.k.a. hierarchical or multilevel nonlinear models—which are workhorse statistical models in the experimental sciences (Fitzmaurice et al., 2008). Examples of modelling problems suitable for NMEMs includes a population of human cells with different combinations of shared proteins, or a population of engineered bacterial cells with different combinations of shared genetic sequences. Through independence assumptions in the probabilistic model, NMEMs separate inference over individual, group, and population-level variability (Karlsson et al., 2015). In VI-HDS, we design block conditioning in the variational distribution to achieve the same hierarchical structure for an arbitrary dynamical system, improving the model fit and predictive accuracy.

We validate our proposed framework with a case study in synthetic biology: modelling the activity of synthetic biological devices in bacteria. We show that VI-HDS has the following desirable properties: (i) performs as well as state-of-the-art methods but is substantially faster to learn, while enabling (ii) scaling to massive datasets by leveraging fast reverse-mode automatic differentiation and amortised inference, (iii) easily incorporates related dynamical systems for efficient learning by sharing statistical strength, (iv) facilitates zero-shot learning through compositional group structure; (v) enables rapid closed-loop iteration across plausible mechanistic models, which can be a mixture of prescribed equations and black-box neural networks.

2. Methodology

In this section we describe our hierarchical and nonlinear dynamical systems model, followed by posterior inference framework using conditional variational autoencoders. We present detailed descriptions of the observable data, followed by the generative process and the variational inference approach. We then discuss how to learn a multilevel generative model alongside a conditional variational posterior. Finally, we demonstrate the flexibility of our approach through the specification of a dynamical system by either prescribed mechanistic equations (*white-box*) or deep neural networks (*black-box*). Our probabilistic assumptions are summarised by the graphical model in Figure 1B.

2.1. Data Model

Repeated observations of groups In many natural sciences it is common to refer to both the observation variables and the state variables as *species*, because they directly or indirectly represent a *kind* or *type* of measurable quantity in the real world, such as protein concentrations. We adopt this convention here. The n th datum $\mathbf{Y}^{(n)}$ is an $M \times T$ matrix, where M is the number of *observed species*, repeatedly measured over T time points. In order to support variability in closely related observations, we introduce the random variable \mathbf{u} to represent such experimental conditions.

Group stratification by composition We model a population of dynamical systems, each containing a different combination of S shared components. A specific combination of components defines a *group*, which we formalise as a multi-hot vector $\mathbf{g} = [\mathbf{c}_1^\top, \dots, \mathbf{c}_S^\top]^\top$, where each \mathbf{c}_s is a one-hot vector that identifies 1 of $K^{(s)}$ components. The multi-hot vector \mathbf{g} has exactly S non-zero entries. By composing groups this way, we can perform *zero-shot learning*, where our model learns to predict the behaviour of unseen groups, as described in Section 2.4 and studied experimentally in Section 5.3.

We now turn to a description of the generative process, which motivates our conditional variational autoencoder inference algorithm.

2.2. Generative Process

We model observations using a latent variable model with the generative process described in Equations 1-5. Latents \mathbf{z} represent both parameters of the dynamical systems model, and, if necessary, noise process parameters (e.g. variances in the likelihood). We emphasise that the structure of $p_\theta(\mathbf{z}|\mathbf{g})$ will depend on whether the model is a white-box, whereby the prior will contain informative domain knowledge, or a black-box, where it will be uninformative diagonal Gaussians.

$$\mathbf{z} \sim p_\theta(\mathbf{z}|\mathbf{g}) \quad (1)$$

$$\dot{\mathbf{x}} = f_\theta(\mathbf{x}; \mathbf{z}, \mathbf{u}, \mathbf{g}) \quad (2)$$

$$\mathbf{X} = \text{Simulate}(f_\theta, \mathbf{x}_0) \quad (3)$$

$$\mathbf{M} = \psi(\mathbf{X}), \quad \Sigma = \rho(\mathbf{X}, \mathbf{z}) \quad (4)$$

$$\mathbf{Y} \sim p(\mathbf{Y}|\mathbf{M}, \Sigma) \quad (5)$$

Central to the generative process is f_θ , the dynamics model describing the behaviour of an individual’s unobserved state vector \mathbf{x} over time, where $\dot{\mathbf{x}}$ denotes the time derivative of \mathbf{x} . Along with latent variables \mathbf{z} and group indicators \mathbf{g} , f depends on \mathbf{u} , a vector of observed *experimental conditions* or *treatments*. Solving the dynamical system results in a state-time matrix \mathbf{X} . The state-time matrix is then mapped to mean \mathbf{M} and, along with \mathbf{z} , covariance Σ of \mathbf{Y} through

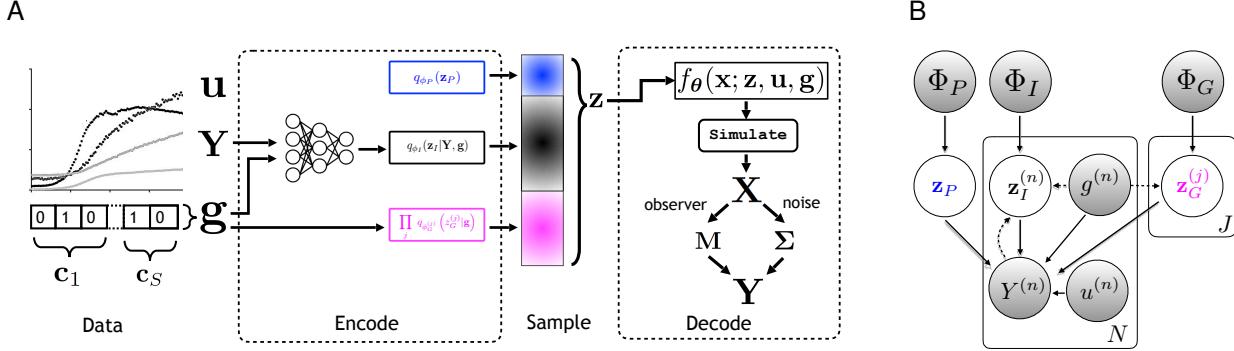


Figure 1. Auto-encoding hierarchical and nonlinear dynamical systems. (A): The computational flow graph for encoding, sampling from the variational posterior, and simulating the dynamical system. Note that the sample and simulate operations are constrained to be differentiable. (B): Probabilistic structure for the sequence-valued dependent variable \mathbf{Y} . Dashed lines represent dependencies in q , and solid lines in p . Φ represents prior information elicited from a domain expert.

an observation process ψ and noise process ρ .

For generality, in (4), we have included observer ψ and noise process ρ functions. For dynamical systems that don't require such processes, ψ can be the identity function and ρ some fixed covariance matrix.

The dynamics model f has parameters θ that are adjusted during learning: for the white-box, they are group-level offsets and scales applied to $\mathbf{z}_G \in \mathbf{z}$; for the black-box they are neural network weights. In all our experiments we assume $p(\mathbf{Y}|\mathbf{M}, \Sigma)$ to be Gaussian. The exact form of the noise process ρ depends on which model we use, and will be explained below according to the white- and black-box framework.

Note that we have written (2) and (3) as a general dynamical system. For processes that exhibit time-varying noise, we can modify (2) and (3) with a time-dependent noise process, to represent the diffusion and drift processes of a stochastic differential equation. However, for clarity of exposition, we will focus on ODE systems as the representative dynamical system in the remainder.

2.3. Auto-Encoding Hierarchical and Nonlinear Dynamical Systems

Core to our approach is amortised stochastic variational inference (Hoffman et al., 2013) with a reparameterised variational distribution (Kingma & Welling, 2013; Rezende et al., 2014), a.k.a. a variational auto-encoder (VAE). We extend this basic framework to hierarchical structure in the dynamical system through conditional variational autoencoders (Sohn et al., 2015). We assume a conditional prior $p_{\theta}(\mathbf{z}|\mathbf{g})$, conditional generative $p_{\theta}(\mathbf{Y}|\mathbf{z}, \mathbf{g}, \mathbf{u})$, and conditional variational $q_{\phi}(\mathbf{z}|\mathbf{Y}, \mathbf{g}, \mathbf{u})$ distributions whose parameters are directly or indirectly calculated using neural networks. Figure 1A illustrates the computational flow of our framework.

Neural networks weights ϕ and θ are adjusted by stochastic gradient ascent to maximise the evidence lower bound (ELBO) on the conditional log marginal data likelihood $\log p(\mathbf{Y}|\mathbf{g}, \mathbf{u})$:

$$\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{Y}, \mathbf{g}, \mathbf{u})} [\log p_{\theta}(\mathbf{Y}|\mathbf{z}, \mathbf{g}, \mathbf{u}) + \log p_{\theta}(\mathbf{z}|\mathbf{g})] - \log q_{\phi}(\mathbf{z}|\mathbf{Y}, \mathbf{g}) \quad (6)$$

VAE training thus simultaneously learns an amortised inference model q and a generative model p , transforming traditional inference over an intractable integral into a gradient-based optimisation procedure, bringing significant computational and scaling benefits to learning and inference. We note that this inference method can also be applied to “simple” mechanistic models (i.e. a fixed decoder model) using likelihood-free inference (Moreno et al., 2016; Tran et al., 2017).

2.4. Multilevel Block-Conditional Encoder

Our approximate posterior $q_{\phi}(\mathbf{z}|\mathbf{Y}, \mathbf{g})$ is factorised into three blocks of independent latent variables representing population, group, and individual factors:

$$q_{\phi_P}(\mathbf{z}_P) \underbrace{q_{\phi_I}(\mathbf{z}_I|\mathbf{Y}, \mathbf{g})}_{\text{Individual}} \underbrace{\prod_j q_{\phi_G^{(j)}}(\mathbf{z}_G^{(j)}|\mathbf{g})}_{\text{Group}}$$

This three-level hierarchy allows us to match latent variable capacity to individual, group, and population level variability according to our model of the data generating process. For example, if the majority of the variability occurs at the group level, more capacity in the variational posterior can be assigned to the group-level block through additional dimensions in \mathbf{z}_G . Note this flexibility generally applies only to the black-box model, since the white-box parameter hierarchy is prescribed by domain knowledge. Next, we formalise the notion of encoding variability at multiple hierarchical

levels. Note that we make a mean-field assumption for the variational posterior, but in general, any distributions may be chosen.

Population-level latent variables are independent of observations and groups. For each of the L dimensions of the population block of the mean-field posterior, we set

$$q_{\phi_P(l)} = \mathcal{N}(\mu = r_l, \sigma^2 = \exp\{2v_l\}),$$

where $\phi_P(l) = \{r_l, v_l\}$, and $r_l, v_l \in \mathbb{R}$.

Group-level latent variables are independent of \mathbf{Y} but dependent on group membership. Recall that a group is a collection of components. As described in section 2.1, we use the multi-hot \mathbf{g}_j to form a summation of group-specific scalars, each of which corresponds to a component in the following way:

$$q_{\phi_G^{(j)}(m)} = \mathcal{N}(\mu = \boldsymbol{\nu}_m^\top \mathbf{g}_j, \sigma^2 = \exp\{2\boldsymbol{\eta}_m^\top \mathbf{g}_j\}),$$

where $\phi_G(m) = \{\boldsymbol{\nu}_m, \boldsymbol{\eta}_m\}$, sharing parameters among the groups through the selector mechanism \mathbf{g}_j .

Individual-level latent variables are encoded in a manner similar to typical VAEs. The weights and biases of a deep neural network are contained in ϕ_I . The networks encode the mean m_k and variance v_k functions for the k 'th factor, so that

$$q_{\phi_I(k)} = \mathcal{N}(\mu = m_k(\mathbf{Y}, \mathbf{g}), \sigma^2 = v_k(\mathbf{Y}, \mathbf{g})).$$

The encoder neural networks perform amortised inference that is optimised during learning.

Alternatively, individual parameters can be optimised per data instance, giving tighter ELBOs. However, non-amortised stochastic variational inference (Hoffman et al., 2013) would not scale with massive amounts of data—one of the key desiderata for our method.

2.5. Block-Conditional Decoder

The bulk of the multilevel modelling occurs in the variational posterior, whose draws are used as parameters in the ODE model f . However, group-conditional modifications are made to \mathbf{z}_G in both the white-box and black-box decoder in order to induce the correct conditional probabilistic structure for the conditional ELBO. For example, before passing $\mathbf{z}_G^{(j)}$ into f , we apply weights $\mathbf{w} \in \theta$, modifying $\mathbf{z}_G^{(j)} \leftarrow \mathbf{z}_G^{(j)} * \exp(w_1^\top \mathbf{g}) + w_2^\top \mathbf{g}$, corresponding to a group-dependent Gaussian; this implies the correct conditional VAE for group-conditional variables*. The specific form of the block-conditional decoder depends on white- or

* Note that in many cases, variables are constrained to be positive, which we model as log-normals. In this case, $\mathbf{z}_G^{(j)}$ is implicitly a group-dependent log-normal.

black-box right-hand side choices for the dynamical system. We discuss these choices next and show how to induce the appropriate multilevel probabilistic structure in each.

Black-Box Dynamics. To simplify notation, let $\Psi = \{\mathbf{z}_P, \mathbf{z}_G, \mathbf{z}_I, \mathbf{u}, \mathbf{g}\}$ and a function based on a neural network be ω . We can write an ODE equation as $\dot{\mathbf{x}} = \omega(\mathbf{x}, \Psi; \theta)$, where the dot indicates the first time derivative. In many cases, including our case study, prior domain knowledge regarding the functional form of the ODE can be incorporated into the black-box. For example, the following ODE is the difference between a growth and degradation term for the latent process, but we know that degradation will go to 0 when $\mathbf{x} \rightarrow 0$. To encode this we use a softplus activation over two networks, denoted by +:

$$\dot{\mathbf{x}} = \omega_1^+(\mathbf{x}, \Psi) - \mathbf{x} \odot \omega_2^+(\mathbf{x}, \Psi) \quad (7)$$

where \odot is the Hadamard product.

Constant and Time-Varying Noise Models. Likelihood $p(\mathbf{Y}|\mathbf{M}, \Sigma)$ requires defining a noise process $\Sigma = \rho(\mathbf{X}, \mathbf{z})$. For the white-box model, we assume *constant variance* for each signal in \mathbf{Y} ; these are encoded using log-normal population-level variables. For the black-box model, we take a non-parametric approach and learn a time-dependent variance model, using an additional black-box ODE model:

$$\dot{\mathbf{v}} = \omega_3^+(\mathbf{v}, \mathbf{x}, \Psi) - \mathbf{v} \odot \omega_4^+(\mathbf{v}, \mathbf{x}, \Psi) \quad (8)$$

In experiments, we solve \mathbf{v} simultaneously with \mathbf{x} , which adds little computational overhead, but provides a flexible, data-driven approach to modelling observation noise. This time-dependent variance process is applied element-wise as Σ in Eq. 5.

2.6. Simulation: Modified Euler

For ODEs, we can explicitly write a solver to simulate the dynamical system. Following (Chen et al., 2018), if the ODE has too many parameters or the discretisation appears to affect the quality of the solution, we can learn gradients in a memory-efficient way through the adjoint method (Stapor et al., 2018). The same simulation applies to white-box and black-box methods, and the difference is contained in the function f_θ .

3. Prior Work

Prior work on learning the parameters of dynamical systems is extensive; we present an overview of key works here. See Appendix A for an extended discussion.

Markov-Chain Monte Carlo (MCMC) methods have long been the gold standard for inference in ODEs (e.g., Xun et al. 2013). Dalchau et al. (2019) applied MCMC inference

to the synthetic biological problem described in Section 4, however reports convergence times of 24-48 hours. Due to numerical integration at each time step, MCMC is not scalable across large time series models, and lacks the flexibility, interpretability, and compositionality of our method.

Likelihood-free methods (a.k.a. Approximate Bayesian Computation (ABC)) can infer parameters of dynamical systems without running a Markov chain to convergence. For instance, Sequential Monte Carlo ABC simulates a discretised dynamical system to infer parameters of an ODE Toni et al. (2009). We choose instead to take advantage of fast, gradient-based optimization that scales better with data.

Gradient matching avoids numerical integration by using a Gaussian Process (GP) regression to the state variables of the dynamical system. Gorbach et al. (2017) use mean-field variational inference and gradient matching to learn parameters GPs that approximate dynamical systems. GPs do not scale well to large datasets, and so are not applicable to our massive data regime.

Variational inference for dynamical systems has been applied heavily in state-space models, e.g. Archer et al. (2015). Unlike (Archer et al., 2015), our method deals with dynamical systems that have hierarchical latent structure and highly *nonlinear* latent transitions. Moreover, our model does not require full state observability, which is infeasible in many applications, including our case study.

Krishnan et al. (2015) learn *nonlinear Kalman filters* through stochastic variational inference. They explicitly generalise linear dynamical systems, discovering arbitrarily complex transition dynamics and emission distributions. A key difference is that the mean and covariance functions for their (tridiagonal) variational distribution are recurrent neural networks.

Ryder et al. (2018) uses variational inference for *stochastic differential equations*. Their approximate posterior factorizes into one component that determines the parameters for the SDE drift and diffusion matrices, and one that predicts the parameters of the latent process. The component that describes the latent process evolution is autoregressive, requiring more computation. Our method induces hierarchical structure in the approximate posterior rather than a complex noise process, and deals with dynamical systems in greater generality. Thus, given the generative process of an SDE, our method could also be used for Bayesian inference over such parameters.

Chen et al. (2018) propose *neural ordinary differential equations* and reinterpret deep neural networks that exhibit repeated composition as discretised ODEs. They use a different modelling regime to ours, in that we learn a block-conditional variational distribution over the parameters of a system of ODEs, while their variational distribution is

over the initial state of the latent time series, with the goal of reinterpreting deep learning models as continuous-time dynamical systems. We work from the opposite direction, inducing hierarchical structure in the variational distribution for more efficient sharing of statistical strength during parameter identification.

4. Synthetic Biology Case Study

We used our methodology to model bacterial cells that were genetically engineered to respond to specific input signals. Cells are typically engineered by inserting DNA that instructs the cells to produce proteins, which can implement a range of information processing behaviours, including Boolean logic gates (Nielsen et al., 2016), analog computation (Daniel et al., 2013), or inter-cellular communication (Grant et al., 2016).

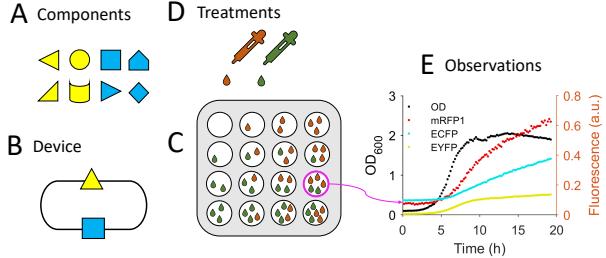


Figure 2. Schematic view of the Synthetic Biology Case Study
A: Exchangeable genetic components are selected and inserted into a circular piece of DNA called a *plasmid* (**B**), which genetically encodes a *receiver device*. Each receiver device contains a different combination of two types of components (yellow and blue), each with several variants. **C:** The plasmid is inserted into a bacterial cell culture, which is replicated and grown in wells (circles) on an experimental plate. **D:** Each well is treated with different concentrations of chemicals C_6 and C_{12} and grown for approximately 24 hours. **E:** During the growing period cameras capture OD, RFP, YFP, CFP.

We considered a population of bacterial cell cultures, each with a different combination of shared genetic components (Figure 2). Each cell culture contained genetic components implementing a *double receiver* device, which functions as a biosensor by producing two *receiver proteins*, R and S . Receiver R binds to input signal C_6 to activate production of cyan fluorescent protein (CFP), and receiver S binds to input signal C_{12} to activate production of yellow fluorescent protein (YFP) (Grant et al., 2016). Genetic components associated with the efficiency of R production were one of (Pcat, RS100, R33) and equivalently one of (Pcat, S32, S175, S34) for S . Each culture continuously produces a red fluorescent protein (RFP), providing a baseline for cellular activity. Finally, the size of the culture is measured as optical density (OD).

White-box model The white-box ODE model we consider describes the evolution of the density c of the cell culture as $\dot{c} = \gamma(c).c$. The function $\gamma(c)$ defines a lag-logistic scheme, considering the growth rate to be 0 before a time t_{lag} is reached, at which point there is a transition to logistic (density-limited exponential) growth.

$$\gamma(c) = r \cdot (1 - c/K) / (1 + e^{-4(t-t_{\text{lag}})}) \quad (9)$$

Additional state variables describe the intracellular concentrations (denoted $[.]$) of receiver device components RFP, YFP, CFP, LuxR (R) and LasR (S), and two additional variables quantifying cellular autofluorescence at 480 nm (F_{480}) and 530 nm (F_{530}).

$$\begin{aligned} \dot{[\text{RFP}]} &= r_c - (\text{d}_{\text{RFP}} + \gamma) \cdot [\text{RFP}] \\ \dot{[\text{CFP}]} &= \text{a}_{\text{CFP}} \cdot r_c \cdot f_{76}(C_6, C_{12}, [R], [S]) - (\text{d}_{\text{CFP}} + \gamma) \cdot [\text{CFP}] \\ \dot{[\text{YFP}]} &= \text{a}_{\text{YFP}} \cdot r_c \cdot f_{81}(C_6, C_{12}, [R], [S]) - (\text{d}_{\text{YFP}} + \gamma) \cdot [\text{YFP}] \\ \dot{[R]} &= \text{a}_R \cdot r_c - (\text{d}_R + \gamma) \cdot [R] \\ \dot{[S]} &= \text{a}_S \cdot r_c - (\text{d}_S + \gamma) \cdot [S] \\ \dot{[F_{480}]} &= \text{a}_{480} \cdot r_c - \gamma \cdot [F_{480}] \\ \dot{[F_{530}]} &= \text{a}_{530} \cdot r_c - \gamma \cdot [F_{530}] \end{aligned}$$

Here, f_{76} and f_{81} are the derived functional responses to input signals and receiver proteins, as defined in Grant et al. (2016) (also see Appendix A.1). The parameter r_c represents the metabolic activity of the culture, which can vary between cultures, and is therefore an individual parameter. We assume that the growth parameters are also specific to a given culture, and so $\mathbf{z}_I = \{r, K, t_{\text{lag}}, r_c\}$. The receiver devices in each culture differed by two genetic components, a_R and a_S , responsible for the production of the two receiver proteins R and S , respectively, such that $\mathbf{z}_G = \{a_R, a_S\}$. All remaining parameters \mathbf{z}_P are shared across all cell cultures. The experiment conditions $\mathbf{u} = [C_6, C_{12}]$ are assumed to be constant throughout the course of an experiment.

Observer process The observer process ψ defines how to relate the model to the ($M = 4$) measured signals. Optical density (OD) represents the density of the culture, and is modelled explicitly by the white-box model. By analogy, we impose that the first internal state of the black-box model (x_0) models cell density. Bulk fluorescence measurements of the whole culture are then collected at excitation wavelengths corresponding to RFP, YFP and CFP. In the white-box model, contributions to bulk fluorescence includes fluorescent proteins and autofluorescence as separate variables; scaling the sum of the protein and autofluorescence concentrations by the cell density provides an approximation of bulk fluorescence for each signal. However, in the black-box model, internal states cannot be separated easily under

addition, so we use single variables each to represent intracellular RFP, YFP and CFP, and do not explicitly separate autofluorescence. Accordingly, the observer processes for each model differ in structure slightly (Table 1).

Table 1. The observer process for white-box and black-box models.

Signal	White-box	Black-box
OD	c	x_0
RFP	$c \cdot [\text{RFP}]$	$x_0 \cdot x_1$
YFP	$c \cdot ([\text{YFP}] + [F_{530}])$	$x_0 \cdot x_2$
CFP	$c \cdot ([\text{CFP}] + [F_{480}])$	$x_0 \cdot x_3$

5. Experiments

We performed two experiments on data from the synthetic biology case study, where measurements of six genetic devices (Pcat-Pcat, RS100-S32, RS100-S34, R33-S32, R33-S175 and R33-S34) were combined into a collection of 312 time-series. In the first experiment, we split the dataset into cross-validation folds and compared a block-conditional white-box model (see Section 4) with a black-box model using biologically plausible constraints on its functional form (see Section 2.5). In the second experiment, the combined dataset had one device held out as a test dataset, while the remaining devices underwent cross-validation for model selection. This experiment was performed on devices whose components had not appeared in combination together in the training set, but had appeared separately in other devices, enabling their parameters to be identified. The ability of the method to predict devices not previously seen before is particularly compelling when seeking to select components that, when put together, implement a given specification.

5.1. Architecture and Optimisation Details

For $q(\mathbf{z}_I | \mathbf{Y}, \mathbf{g})$, we use the same encoder NN for both white-box and black-box models: 10 1D convolutional filters, feeding 50 unit hidden layer with tanh activations; \mathbf{g} is concatenated to the hiddens, which is then connected to the mean and variances outputs of q . Neural networks ω^+ all have 25 hidden units. All decoder NNs are the same for all black-box models. During training we used a K=100 importance weighted auto-encoder (IWAE) estimator for gradient computation (Burda et al., 2015). We implemented a doubly reparameterised gradient estimator (DReG) for IWAE, which has recently been shown to yield lower-variance gradient estimates than the gradient computed naively by reverse-mode automatic differentiation (Tucker et al., 2018), while also avoiding the known problem of poor gradients for the variational approximation with greater numbers of importance samples (Rainforth et al., 2018). We found that the DReG estimator gave modest improvements for the black-

box, but *very significant improvements* in performance and rate of convergence for the white-box (see Appendix figure 6). For evaluation, we used $K = 1000$ IWAE samples. We implemented a modified-Euler solution to simulate our ODEs; this was implemented in tensorflow using tf.while blocks. We ran all experiments for 500 epochs using Adam optimisation (Kingma & Ba, 2014). To perform 4-fold cross-validation, black-box models take approximately 40 minutes and white-box models approximately 2 hrs. This is significantly faster than MCMC (see Section ??), which required approximately 40 times more computation.

5.2. Multiple Device Inference

We first ran 4-fold cross-validation with 500 epochs and batch size 36, and then collected the posterior predictive distributions for the cross-validated portions of each fold. Both white-box and black-box models were able to capture the majority of the dynamical behaviour (Figures 3, 7–13). In Table 2 cross-validation ELBO estimates are presented for variety of latent variable and ODE signal capacity for the black-box. Since the encoder neural networks are fixed for both models, we only report the single estimated ELBO for the white-box. Black-box has two significant advantages that explain the improved performance: 1) it has flexible modelling capacity in the form of its neural differential equations, and 2) it has a more suitable and realistic neural noise process, enabling the posterior predictive distribution to exhibit lower variance, especially at earlier time points. In Figure 4C we plot the dynamic noise process for the black-box model.

Table 2. Black-box Cross-validation Results. For comparison, the white-box cross-validation ELBO is 1047 nats. M_x is the number of species in the black-box, where the first 4 are used in the observer process, so $M_x = 5$ means there is 1 extra *latent species*. For the figures we select the black-box model *, but we note no significant difference between results shown *italics*.

# Variables			Mean IWAE ELBO, K=1000			
P	G	I	$M_x = 4$	$M_x = 5$	$M_x = 6$	$M_x = 8$
2	0	0	1155	1215	1210	1213
2	2	0	1201	1277	1280	1283
2	2	2	1175	<i>1352</i>	<i>1340</i>	<i>1340</i>
5	2	5	1204	1360*	1321	1305

To provide a summary view of how each model predicts the response to the input signals C_6 and C_{12} , we plotted the posterior predictive distributions at the final time-points as a function of the input concentrations. For the R33-S175 device, the responses of CFP to C_6 and YFP to C_{12} are both monotonically increasing. These responses are well-predicted by both models (Figures 5A, 14), though the black-box model captured the actual measured values with

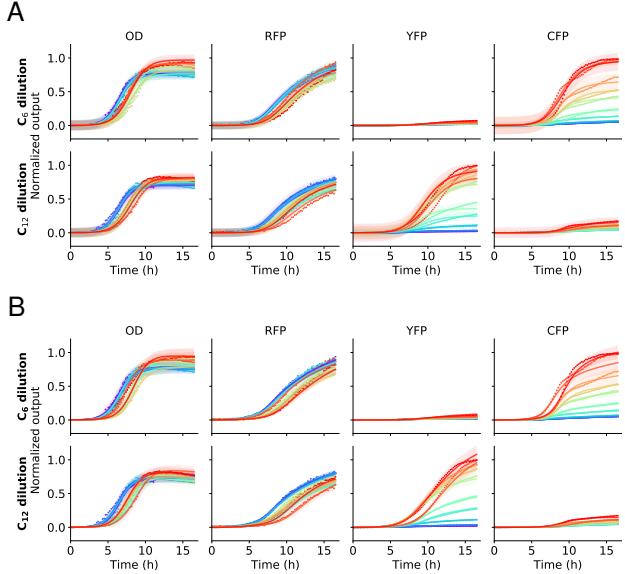


Figure 3. Posterior predictive distribution for the R33-S175 device. (A) White-box model. (B) Black-box model. For clarity, we selected one of each replicated treatment, so that a full dilution series can be visualised for C_6 and C_{12} , sweeping from low to high YFP/CFP expression. Predictions for the full dataset are shown in the Appendix (Figure 7).

higher precision. In contrast, the R33-S34 device exhibits a non-monotonic shape of YFP expression in response to C_{12} . Because the transfer function hypothesised to describe the equilibrium response of YFP production (f_{81} , see Appendix) is monotonic increasing with respect to C_{12} , it would seem that it cannot explain the observed trends for R33-S34 in bulk fluorescence. Here we find that this can be explained by high C_{12} concentrations slowing down cell growth (Figure 12). While the intracellular concentration of YFP is increased by the monotonic f_{81} , lower cell densities produce lower bulk fluorescence. While there is low precision in the predictions at the final time points of high C_{12} treatments for the white-box model, it captures the general trend almost as well as the black-box model. This valuable modelling insight was really made possible by the simultaneous inference of multiple devices and multiple signals.

Using the black-box approach, we can overcome model misspecification, producing accurate predictions across all 6 devices. However, the interpretability of these models is more restricted. Nevertheless, we made an effort to enforce plausibility constraints on the state variables of the black-box model, both through separating production and degradation terms (ensuring non-negativity of the state variables), and using a similar observer process. As such, comparison of the dynamics of the state variables representing intracellular concentrations in the black-box model might inform what is inaccurate in the white-box model. For example, during the

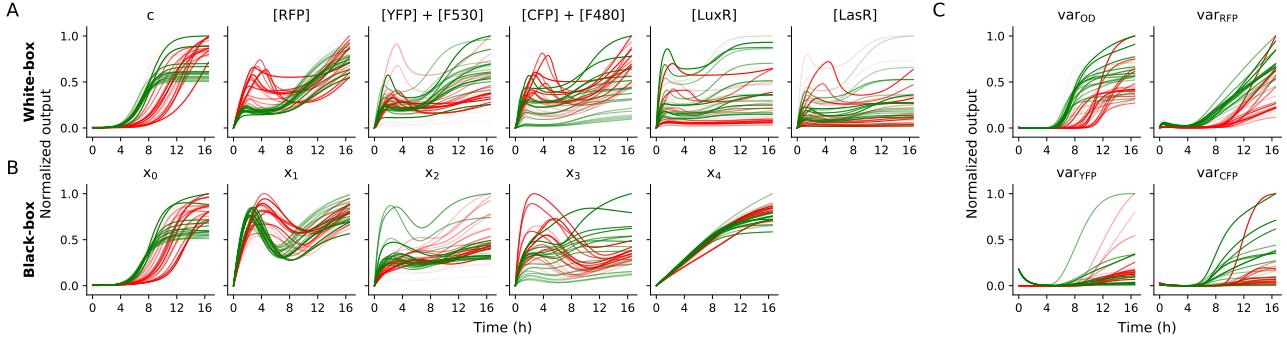


Figure 4. Predicted dynamics of the state variables. The internal state variables for simulation of R33-S34 in the cross-validation experiment with the (A) white-box and (B) black-box models are compared. Simulations are separated by whether they correspond to C_6 treatments (green) or C_{12} treatments (red), with lower concentrations indicated by lighter shades. (C) Dynamics of internal states used for the time-varying noise of the black-box model.

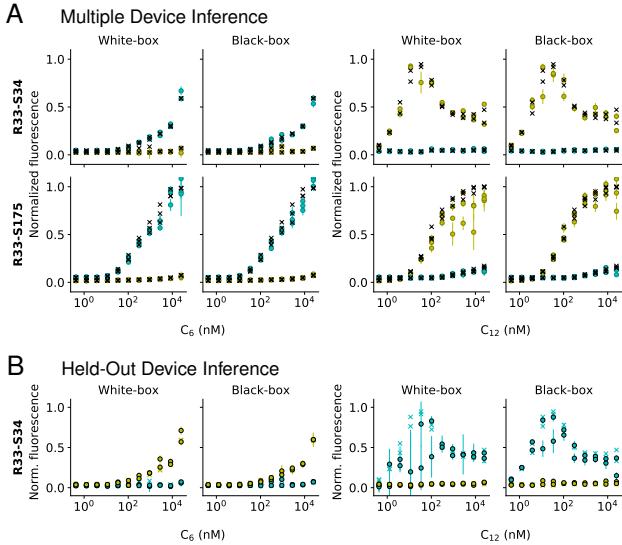


Figure 5. Predicted model response to input signals. (A) 4-fold cross-validation inference and (B) Held-out device inference. To summarise how each model predicts the overall response of R33-S34 and R33-S175 to input signals, we show the posterior predictive distribution at the final time-point (circle with error bars denoting two standard deviations), compared against the data point (crosses) for YFP (yellow) and CFP (cyan).

first 4 hours following treatment by C_6/C_{12} , the white-box model has more accentuated transient variations in [YFP] and [CFP] (Figures 4A, 15) than equivalent state variables in the black-box model (Figures 4, 16).

5.3. Held-out Device Inference

We explored the more challenging scenario of predicting data for devices held-out from the training set. When holding out the R33-S34 device, both the white-box and black-box models were able to accurately capture the moderate increasing response of CFP and almost absent response of

YFP to increasing concentrations of C_6 (Figure 5B). The predictions of the YFP response to C_{12} by the white-box model were imprecise, indicating that either more training data is required to make clearer predictions, that over-fitting has occurred, or that inherent model misspecification is limiting the ability to predict accurately. But as the black-box model was able to correctly predict the observed non-monotonic response with moderate precision, we can rule out the lack of training data as a cause. In general, the ability to compare black-box and white-box models in this way is useful for providing an upper bound on what might be achievable by any prescribed model.

6. Discussion and Future Work

We have introduced **VI-HDS**, a framework for learning and inference of hierarchical and nonlinear dynamical models, and applied it to a careful set of experiments on a synthetic biology case study. VI-HDS brings several elements from recent machine learning advances together into one framework with the following key advantages: efficient amortised inference using VAEs, multilevel latent variable modelling with block-conditional factorisation; a broad range of dynamic behaviour modelling, from user prescribed white-box dynamics to black-box dynamics with neural sub-components.

Although the experiments for our synthetic biology case study focused on ODEs—already a broad class of dynamical systems—our framework could be further generalised to include stochasticity in the dynamics model, for both the white-box and black-box cases. We could potentially avoid sampling altogether by adopting deterministic VI (Wu et al., 2019) to propagate the variational posterior all the way through the ODE solver to the likelihood. Treating inference as optimisation opens the door to novel active learning experiments for dynamical systems of synthetic biology, where additional cost functions provide gradients to

maximise desirable properties, similar to work in synthetic chemistry (Gómez-Bombarelli et al., 2018).

Acknowledgments

Geoffrey Roeder is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number PGSD3-518716-2018.

References

- Archer, E., Park, I. M., Buesing, L., Cunningham, J., and Paninski, L. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.
- Balagaddé, F. K., Song, H., Ozaki, J., Collins, C. H., Barnett, M., Arnold, F. H., Quake, S. R., and You, L. A synthetic escherichia coli predator-prey ecosystem. *Molecular systems biology*, 4(1):187, 2008.
- Becker-Ehmck, P., Peters, J., and van der Smagt, P. Switching linear dynamics for variational bayes filtering. 2018.
- Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.
- Calderhead, B. and Girolami, M. Estimating bayes factors via thermodynamic integration and population mcmc. *Computational Statistics & Data Analysis*, 53(12):4028–4045, 2009.
- Calderhead, B. and Girolami, M. Statistical analysis of nonlinear dynamical systems using differential geometric sampling methods. *Interface focus*, pp. rsfs20110051, 2011.
- Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.
- Chen, Y., Kim, J. K., Hirning, A. J., Josić, K., and Bennett, M. R. Emergent genetic oscillations in a synthetic microbial consortium. *Science*, 349(6251):986–989, 2015.
- Cumbers, J. and Schmieder, K. *What’s Your Bio Strategy: How to Prepare Your Business for Gene Editing and Synthetic Biology*. Pulp Bio Books, 2017. ISBN 9780999313619. URL <https://books.google.co.uk/books?id=SLUbAEACAAJ>.
- Dalchau, N., Grant, P. K., Vaidyanathan, P., Gravill, C., Spaccasassi, C., and Phillips, A. Scalable dynamic characterization of synthetic gene circuits. *bioRxiv*, pp. 635672, 2019.
- Daniel, R., Rubens, J. R., Sarpeshkar, R., and Lu, T. K. Synthetic analog computation in living cells. *Nature*, 497(7451):619–623, May 2013.
- Davidian, M. and Giltinan, D. M. Nonlinear models for repeated measurement data: an overview and update. *Journal of agricultural, biological, and environmental statistics*, 8(4):387, 2003.
- Elowitz, M. B. and Leibler, S. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, jan 2000. ISSN 00280836. doi: 10.1038/35002125. URL <http://dx.doi.org/10.1038/35002125> <http://www.nature.com/doifinder/10.1038/35002125>.
- Ferrell, J. E., Tsai, T. Y. C., and Yang, Q. Modeling the cell cycle: Why do certain circuits oscillate? *Cell*, 144(6):874–885, 2011. ISSN 00928674. doi: 10.1016/j.cell.2011.03.006. URL <http://dx.doi.org/10.1016/j.cell.2011.03.006>.
- Fitzmaurice, G., Davidian, M., Verbeke, G., and Molenberghs, G. *Longitudinal data analysis*. CRC press, 2008.
- Gao, Y., Archer, E. W., Paninski, L., and Cunningham, J. P. Linear dynamical neural population models through nonlinear embeddings. In *Advances in neural information processing systems*, pp. 163–171, 2016.
- Gardner, T. S., Cantor, C. R., and Collins, J. J. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(6767):339–342, jan 2000. ISSN 0028-0836. doi: 10.1038/35002131. URL <http://www.ncbi.nlm.nih.gov/pubmed/10659857> <http://www.nature.com/doifinder/10.1038/35002131>.
- Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.
- Gorbach, N. S., Bauer, S., and Buhmann, J. M. Scalable variational inference for dynamical systems. In *Advances in Neural Information Processing Systems*, pp. 4806–4815, 2017.
- Grant, P. K., Dalchau, N., Brown, J. R., Federici, F., Rudge, T. J., Yordanov, B., Patange, O., Phillips, A., and Haseloff, J. Orthogonal intercellular signaling for programmed spatial behavior. *Mol. Syst. Biol.*, 12(1):849, Jan 2016.
- Hodgkin, A. L. and Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–44, aug 1952. ISSN 0022-3751. URL <http://www.ncbi.nlm.nih.gov/pubmed/12991237> <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Search&term=PMC1392413>.
- Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

- Johnson, M. J., Duvenaud, D., Wiltschko, A. B., Datta, S. R., and Adams, R. P. Structured vaes: Composing probabilistic graphical models and variational autoencoders. *arXiv preprint arXiv:1603.06277*, 2, 2016.
- Karl, M., Soelch, M., Bayer, J., and van der Smagt, P. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.
- Karlsson, M., Janzén, D. L., Durrieu, L., Colman-Lerner, A., Kjellsson, M. C., and Cedersund, G. Nonlinear mixed-effects modelling for single cell estimation: when, why, and how to use it. *BMC systems biology*, 9(1):52, 2015.
- Khalil, A. S. and Collins, J. J. Synthetic biology: Applications come of age. *Nature Reviews Genetics*, 11(5):367–379, 2010. ISSN 14710056. doi: 10.1038/nrg2775. URL <http://dx.doi.org/10.1038/nrg2775>.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Krishnan, R. G., Shalit, U., and Sontag, D. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.
- Leloup, J.-C. and Goldbeter, A. Toward a detailed computational model for the mammalian circadian clock. *Proceedings of the National Academy of Sciences*, 100(12):7051–7056, 2003.
- Linderman, S. W., Miller, A. C., Adams, R. P., Blei, D. M., Paninski, L., and Johnson, M. J. Recurrent switching linear dynamical systems. *arXiv preprint arXiv:1610.08466*, 2016.
- Moore, S. J., MacDonald, J. T., Wienecke, S., Ishwarbhai, A., Tsipa, A., Aw, R., Kyliolis, N., Bell, D. J., McClymont, D. W., Jensen, K., et al. Rapid acquisition and model-based analysis of cell-free transcription–translation reactions from nonmodel bacteria. *Proceedings of the National Academy of Sciences*, 115(19):E4340–E4349, 2018.
- Moreno, A., Adel, T., Meeds, E., Rehg, J. M., and Welling, M. Automatic variational ABC. *arXiv preprint arXiv:1606.08549*, 2016.
- Nielsen, A. A., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., Ross, D., Densmore, D., and Voigt, C. A. Genetic circuit design automation. *Science*, 352(6281):aac7341, Apr 2016.
- Potvin-Trottier, L., Lord, N. D., Vinnicombe, G., and Paulson, J. Synchronous long-term oscillations in a synthetic gene circuit. *Nature*, 538(7626):514–517, oct 2016. ISSN 0028-0836. doi: 10.1038/nature19841. URL <http://dx.doi.org/10.1038/nature19841><http://www.nature.com/doifinder/10.1038/nature19841>.
- Rainforth, T., Kosiolek, A. R., Le, T. A., Maddison, C. J., Igl, M., Wood, F., and Teh, Y. W. Tighter variational bounds are not necessarily better. *arXiv preprint arXiv:1802.04537*, 2018.
- Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- Ruder, W. C., Lu, T., and Collins, J. J. Synthetic Biology Moving into the Clinic. *Science*, 333(6047):1248–1252, sep 2011. ISSN 0036-8075. doi: 10.1126/science.1206843. URL <http://www.sciencemag.org/cgi/doi/10.1126/science.1206843>.
- Ryder, T., Golightly, A., McGough, A. S., and Prangle, D. Black-box variational inference for stochastic differential equations. *arXiv preprint arXiv:1802.03335*, 2018.
- Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pp. 3483–3491, 2015.
- Stapor, P., Froehlich, F., and Hasenauer, J. Optimization and uncertainty analysis of ode models using second order adjoint sensitivity analysis. *bioRxiv*, pp. 272005, 2018.
- Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.
- Tran, D., Ranganath, R., and Blei, D. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pp. 5523–5533, 2017.
- Tucker, G., Lawson, D., Gu, S., and Maddison, C. J. Doubly reparameterized gradient estimators for monte carlo objectives. *arXiv preprint arXiv:1810.04152*, 2018.
- Tyson, J. J., Chen, K., and Novak, B. Network dynamics and cell physiology. *Nature Reviews Molecular Cell Biology*, 2(12):908–916, 2001. ISSN 14710072. doi: 10.1038/35103078.
- Woods, M. L., Leon, M., Perez-Carrasco, R., and Barnes, C. P. A statistical approach reveals designs for the most robust stochastic gene oscillators. *ACS synthetic biology*, 5(6):459–470, 2016.

Wu, A., Roy, N. G., Keeley, S., and Pillow, J. W. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. In *Advances in Neural Information Processing Systems*, pp. 3496–3505, 2017.

Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernandez-Lobato, J. M., and Gaunt, A. L. Deterministic variational inference for robust Bayesian neural networks. To appear in *International Conference of Learning Representations*, 2019.

Xun, X., Cao, J., Mallick, B., Maity, A., and Carroll, R. J. Parameter estimation of partial differential equation models. *Journal of the American Statistical Association*, 108(503):1009–1020, 2013.

Appendix

A. Extended Case Study Description

The synthetic genetic circuits we have used in this work are built from gene cassettes comprised of DNA sequences encoding a promoter, ribosome binding site, coding region, and terminator. These cassettes are assembled into plasmids that are used to transform *E. coli* cells. We refer to a collection of cassettes that implement a particular design as a *device*. The devices used in this paper are all *double receiver* devices that respond to 3-oxo-C6-homoserine lactone (C6) by producing cyan fluorescent protein (CFP) and to 3-oxo-C12-homoserine lactone (C12) by producing yellow fluorescent protein (YFP) (Grant et al., 2016). These devices are built of 5 cassettes. The first 4 cassettes are arranged on a plasmid, which includes one cassette each for producing luxR and lasR proteins (*R* and *S* in the main text), the receiver proteins that bind C6 and C12, respectively, a CFP cassette activated by C6-bound luxR, and a YFP cassette activated by C12-bound lasR. The fifth cassette is chromosomally integrated, and constitutively expresses RFP. The devices vary in the strength of the ribosome binding sites in the luxR and lasR cassettes, creating devices that vary in the amount of each of those proteins expressed and therefore their sensitivity to C6 and C12.

A.1. White-box (mechanistic)

Our general approach for constructing prescribed (white-box) models of biological circuits combines a population-level model for cell culture growth with more detailed models for the concentrations of intra- and intercellular molecules, and resembles the approach commonly used in the synthetic biology literature (Balagaddé et al., 2008; Daniel et al., 2013; Chen et al., 2015; Dalchau et al., 2019). Cell growth models are generally described by the product of the current cell density $c(t)$ and the *specific growth rate* $\gamma(c(t))$, which describes both the per capita growth rate and the decrease in intracellular concentrations due to an increased volume. As explained in the main text, we used a smoothed version of the lag-logistic model for cell growth here.

To model the cellular biochemistry, we translate chemical reaction networks to ODEs using mass action kinetics, which assumes that reactions fire at a rate proportional to the concentration of the reactants. Translating chemical reactions in this way in general leads to a large number of equations, because all mRNAs, proteins, small molecules and complexes between each produce their own equation. As such, model reductions are commonly applied to reduce the number of dependent variables, but result in more complex nonlinearities. Following this approach, the white-box model we consider here (Section 4) was derived in detail previously

(Grant et al., 2016; Dalchau et al., 2019). It describes the time-evolution of the response of double receiver devices to HSL signals C_6 and C_{12} , vector \mathbf{u} in (2). The latent variables \mathbf{x} in (2) are the culture density c , the intracellular concentrations of each expressed protein (luxR, lasR, RFP, CFP, YFP) and variables for autofluorescence, which we model as concentration of intracellular material fluorescent at 480 nm (F_{480}) and 530 nm (F_{530}).

As there are no mRNA species, the variables a_k describe a lumped maximal rate of transcription and translation. The variables d_k describe the intracellular degradation rates of each protein.

The *response* functions f_{76} and f_{81} describe the inducibility of CFP and YFP to complexes involving the HSL signals and the receiver proteins luxR and lasR. The response functions were derived from chemical reactions, making the assumption that signal-receiver binding and unbinding is faster than protein synthesis and degradation (Dalchau et al., 2019). This results in very complex functions, but they are still interpretable as exhibiting saturation behaviour, which occurs as either receiver proteins or promoters become limiting. We define $B_R^{(k)}$ and $B_S^{(k)}$ as the fractions of luxR and lasR proteins bound by an HSL signal

$$B_R = \frac{(K_{R6} \cdot C_6)^{n_R} + (K_{R12} \cdot C_{12})^{n_R}}{(1 + K_{R6} \cdot C_6 + K_{R12} \cdot C_{12})^{n_R}} \quad (10a)$$

$$B_S = \frac{(K_{S6} \cdot C_6)^{n_S} + (K_{S12} \cdot C_{12})^{n_S}}{(1 + K_{S6} \cdot C_6 + K_{S12} \cdot C_{12})^{n_S}}, \quad (10b)$$

These functions are bounded above by 1, which occurs when luxR/lasR become limiting. The CFP or YFP genes are transcribed more efficiently when their promoters are bound by one of the receiver-signal complexes. As such, an additional saturation can be observed within the derived forms

$$f_j(R, S, C_6, C_{12}) = \frac{\epsilon^{(j)} + K_{GR}^{(j)} R^2 B_R + K_{GS}^{(j)} S^2 B_S}{1 + K_{GR}^{(j)} R^2 B_R + K_{GS}^{(i)} S^2 B_S} \quad (11)$$

where $j \in \{76, 81\}$. Here, the parameters $K_{GR}^{(j)}$ and $K_{GS}^{(j)}$ are the affinity constants of receiver-signal complexes for each promoter j , and $\epsilon^{(j)}$ is the leaky rate of transcription/translation in the absence of an activating complex (such as when there is no HSL).

The specific growth rate $\gamma(c_i)$ describes the per-capita cellular growth rate of culture i . Cultures are subscripted in this way because their growth parameters will be local to the culture, which enables implicit accounting for feedback from circuit activity or extrinsic factors that vary in different experiments. As in (Dalchau et al., 2019), we use a lag-logistic growth model which explicitly quantifies a *lag* phase of bacterial growth before an *exponential* phase and

then *stationary* phase. This is usually formulated as

$$\gamma(c_i) = \begin{cases} r_i \cdot (1 - \frac{c}{K_i}), & t > t_{\text{lag},i} \\ 0, & t < t_{\text{lag},i} \end{cases}$$

but to ensure that the right-hand sides of f are differentiable, we replace the switch at $t_{\text{lag},i}$ with a steep sigmoid (Equation 9).

Finally, we remind the reader that we consider the application of this model to multiple devices, in which different ribosome-binding site sequences have been used for the luxR and lasR genes (Section 4). In this mechanistic model, it is the parameters a_R and a_S that are allowed to be device-conditioned. Therefore, using the one-hot mapping strategy for specifying the rbs elements in each device, the model can take on device-specific quantities for luxR and lasR synthesis.

B. Encoder Architecture

Local random effects. The parameters ϕ_{local} consist of real scalar functions m and s such that

$$q_{\phi_{\text{local}}}^{(i)} = \mathcal{N}(m(\mathbf{y}, \mathbf{u}, \mathbf{o}; \xi_i), s(\mathbf{y}, u, \mathbf{o}; \eta_i))$$

The m and s functions have the following form. In the following explication, take \mathbf{y} and u to be arbitrary but particular samples from the true data-generating distribution.

We first apply a 1D strided CNN to the input \mathbf{y} s.t. $\tilde{\mathbf{y}} = \text{flatten}(\text{AvgPool}(\text{ReLU}(\mathbf{y} * W_1)))$, where the convolution filter W_1 is chosen so that $\mathbf{y} * W_1$ has one output channel. Note that the width of the convolution filter influences the correlation structure we can discover in the time series, acting as a window across the sequence. This stage in the computational pipeline is designed for feature learning.

Next, $\tilde{\mathbf{y}}, u$, and \mathbf{o} are concatenated as inputs to an MLP with \tanh nonlinearity and no bias term, that is, $h = \tanh(W_2 \cdot [\tilde{\mathbf{y}}; u; \mathbf{o}])$. The composition of the two stages is a differentiable map $m := h \circ \tilde{\mathbf{y}}$ s.t. $m : \mathbb{R}^{4T+n_u+n_d} \rightarrow \mathbb{R}$, yielding a mean for the learned posterior distribution over that parameter of the ODE.

The s function is identical to the m function but with different parameters, and it further exponentiates the output of the MLP. This last mapping is equivalent to interpreting the unconstrained output of the second neural network as the log of the standard deviation. We require the final step for m to ensure that the resulting covariance matrix of the diagonal Gaussian is PSD. Each layer of $m^{(i)}$ and $s^{(i)}$ is L2-regularized using independent constants λ_j^i to counter overfitting during learning.

Over minibatch learning, the network parameters that map data and conditioning information to means and log standard

deviations are adapted through gradient descent, achieving amortization and supporting generalization.

Global random effects. The parameters ϕ_{global} are independent of the input data, and are real scalars r and v such that

$$q_{\phi_{\text{global}}}^{(i)} = \mathcal{N}(r_i, \exp\{v_i\}),$$

that is, we optimize v_i in an unconstrained space interpreting it as the log standard deviation. The variables r_i and v_i are initialized to the prior, and then optimized through gradient descent to maximize the ELBO.

The prior was set in consultation with the domain experts who designed and conducted the experiments, collected the data, and defined the system of differential equations to capture reasonable upper and lower ranges of the parameters of interest. We note that this could also have been achieved by empirical Bayes or Type-II Maximum Likelihood, which involves learning a hyperprior over the prior parameters, but have left this to future work given the availability of expert knowledge.

Each $q_{\phi_{\text{global}}}^{(i)}$ is a 1D Gaussian or LogNormal (e.g., its output is exponentiated) depending on whether the term it represents in the ODE is a strictly positive real-world quantity or it represents any real value.

C. Supplementary Figures

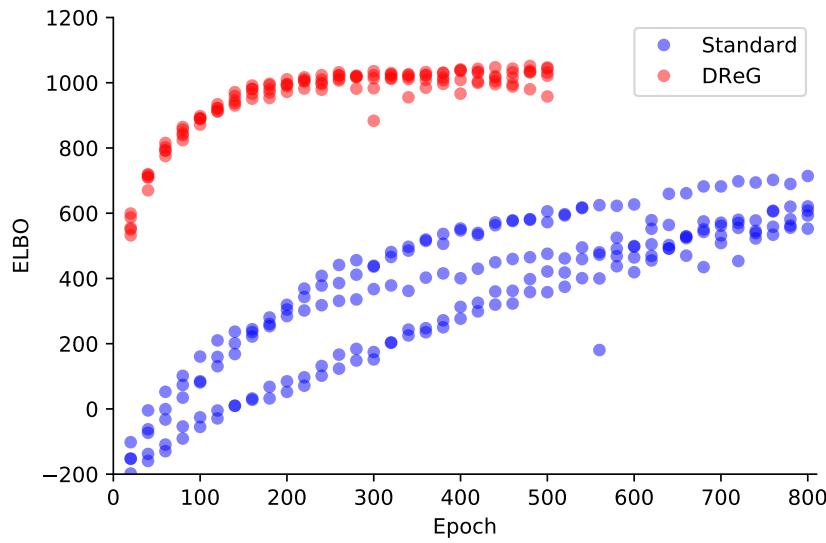


Figure 6. The convergence of the ELBO is improved by the DReG estimator. Shown are 5 independent evaluations of the VI method applied to the prescribed *constant* model, using the standard gradient estimator (blue) and the DReG estimator (red). Each reported ELBO score is the average of a 4-fold cross-validation at a given epoch.

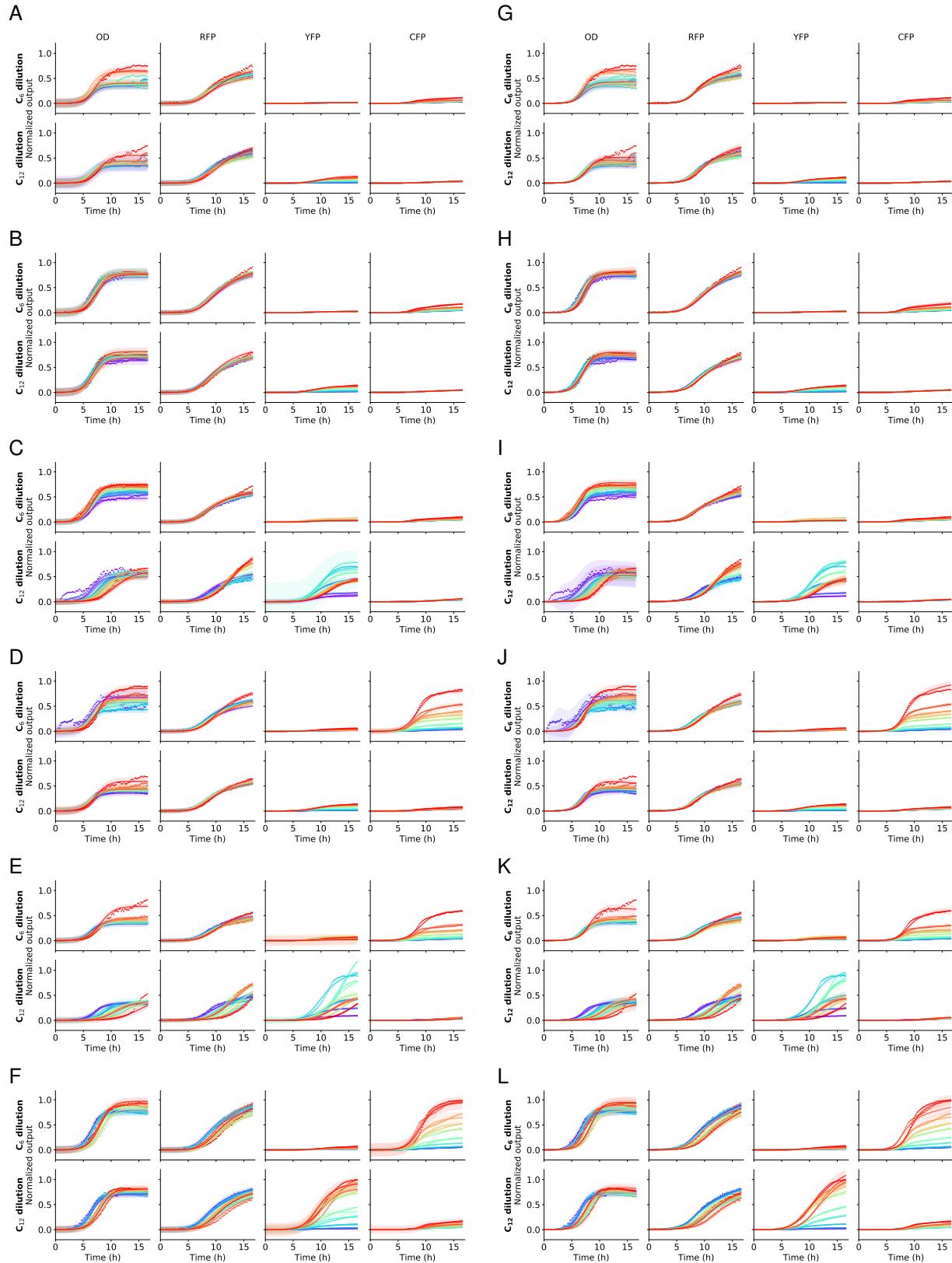


Figure 7. Summary plots for the white-box model. Time-series are partitioned by device and by treatment (C_6 or C_{12}), with a color scale indicating the concentration (warm colors indicate higher concentration). Devices shown are **(A)** Pcat-Pcat, **(B)** RS100-S32, **(C)** RS100-S34, **(D)** R33-S32, **(E)** R33-S34 and **(F)** R33-S175.

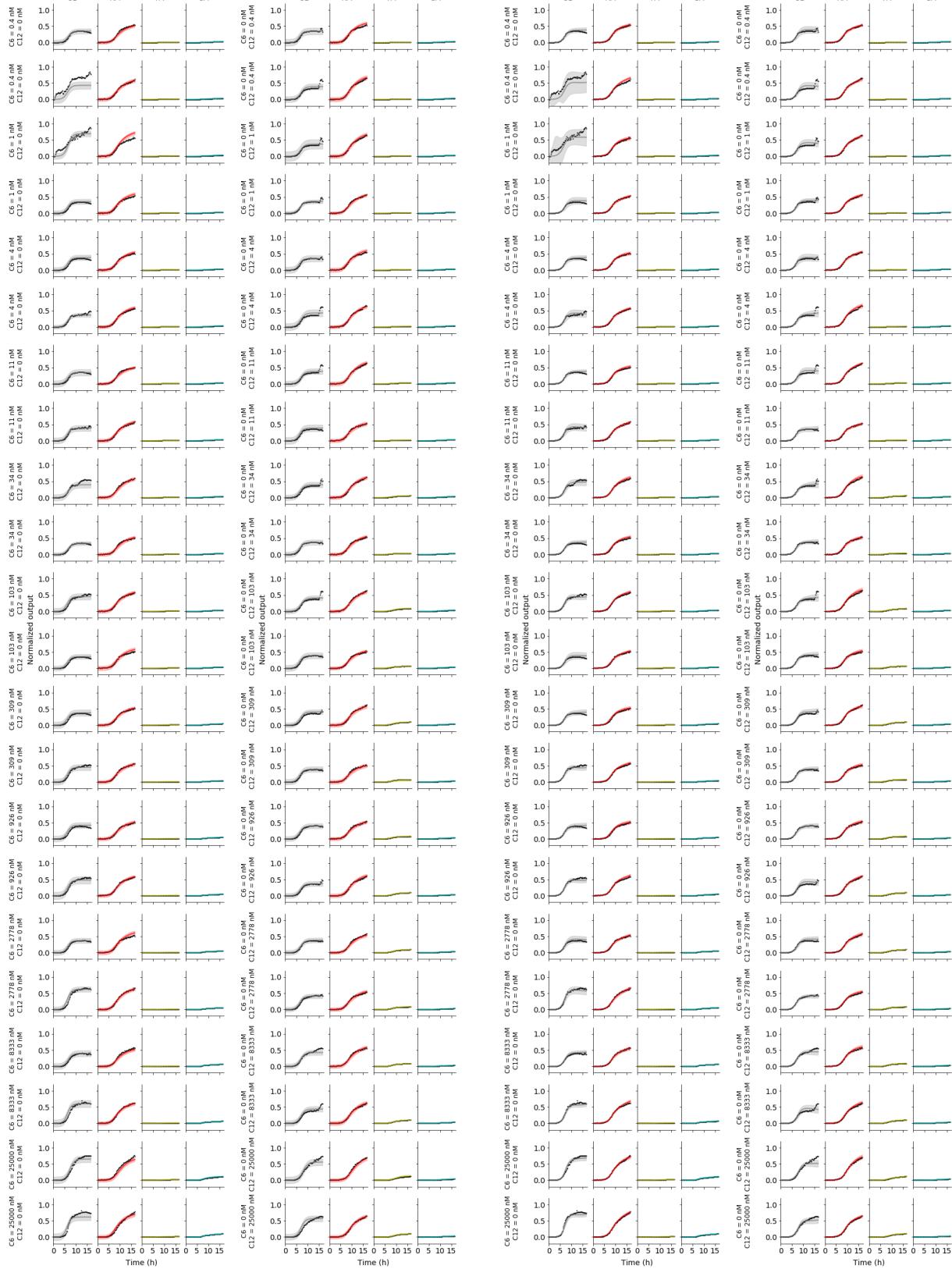


Figure 8. Posterior predictive distribution for 4-fold cross-validation experiment: Pcat-Pcat device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

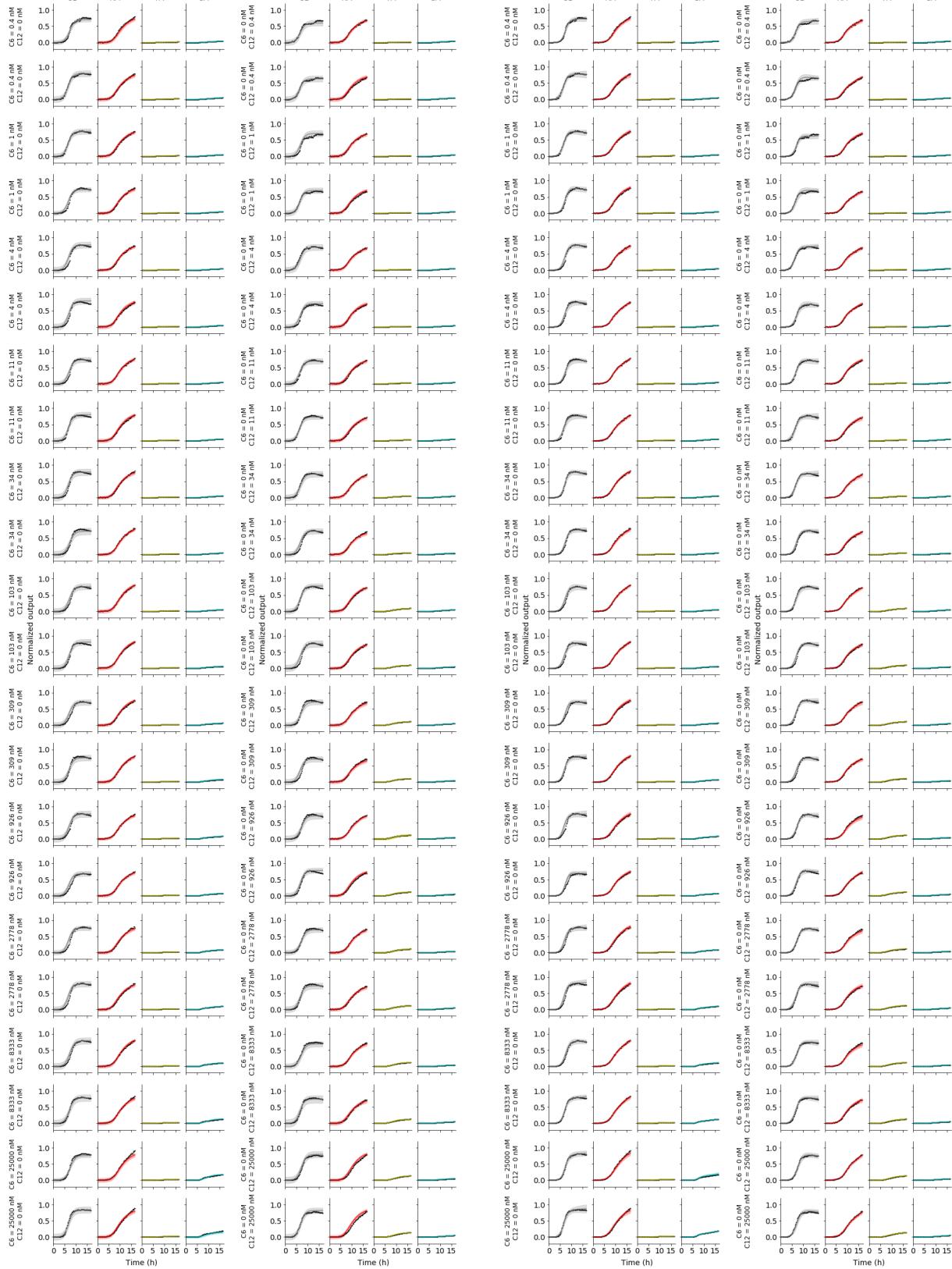


Figure 9. Posterior predictive distribution for 4-fold cross-validation experiment: RS100-S32 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

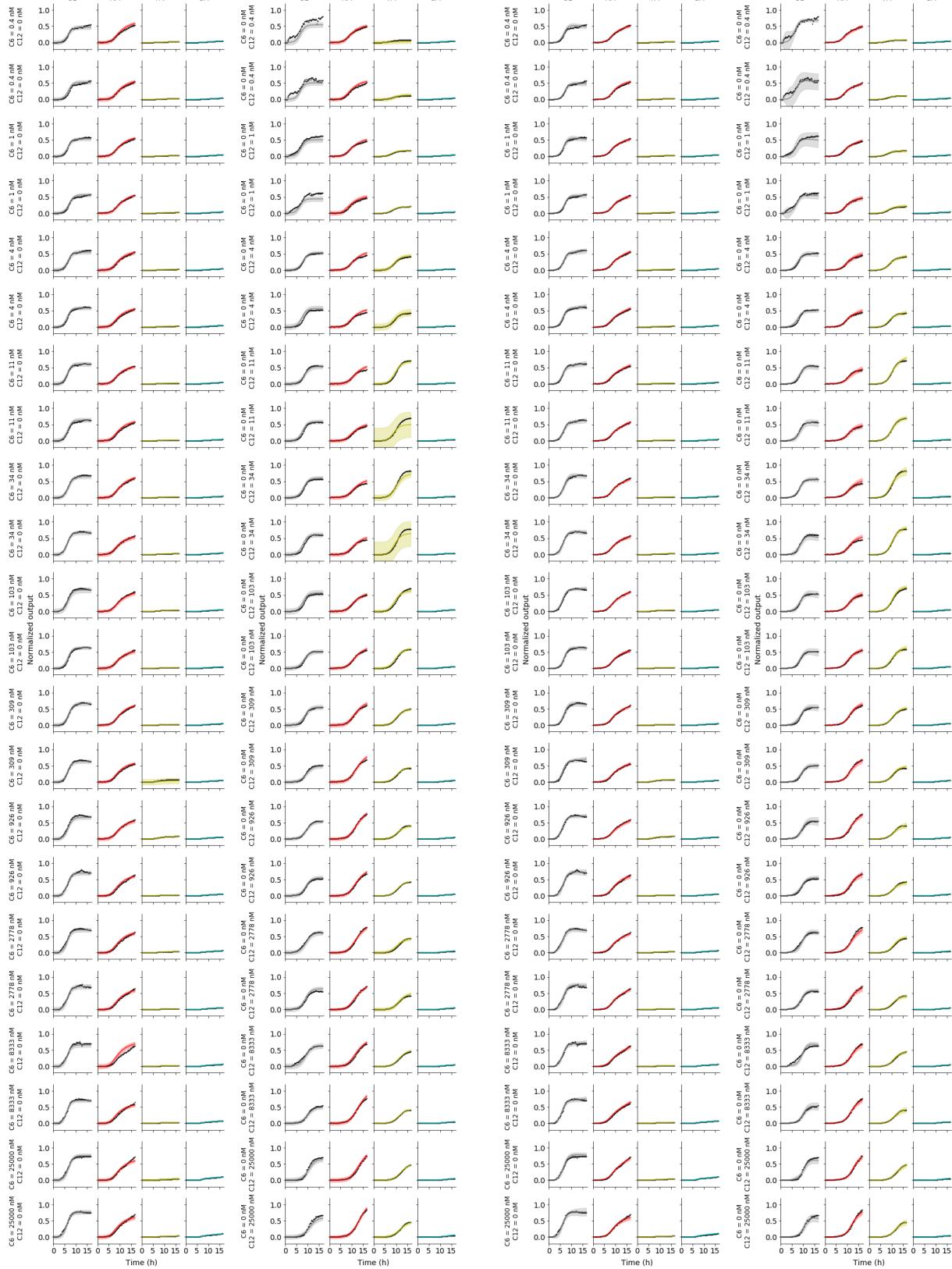


Figure 10. Posterior predictive distribution for 4-fold cross-validation experiment: RS100-S34 device. The left half shows the white-box model, and the right half shows black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

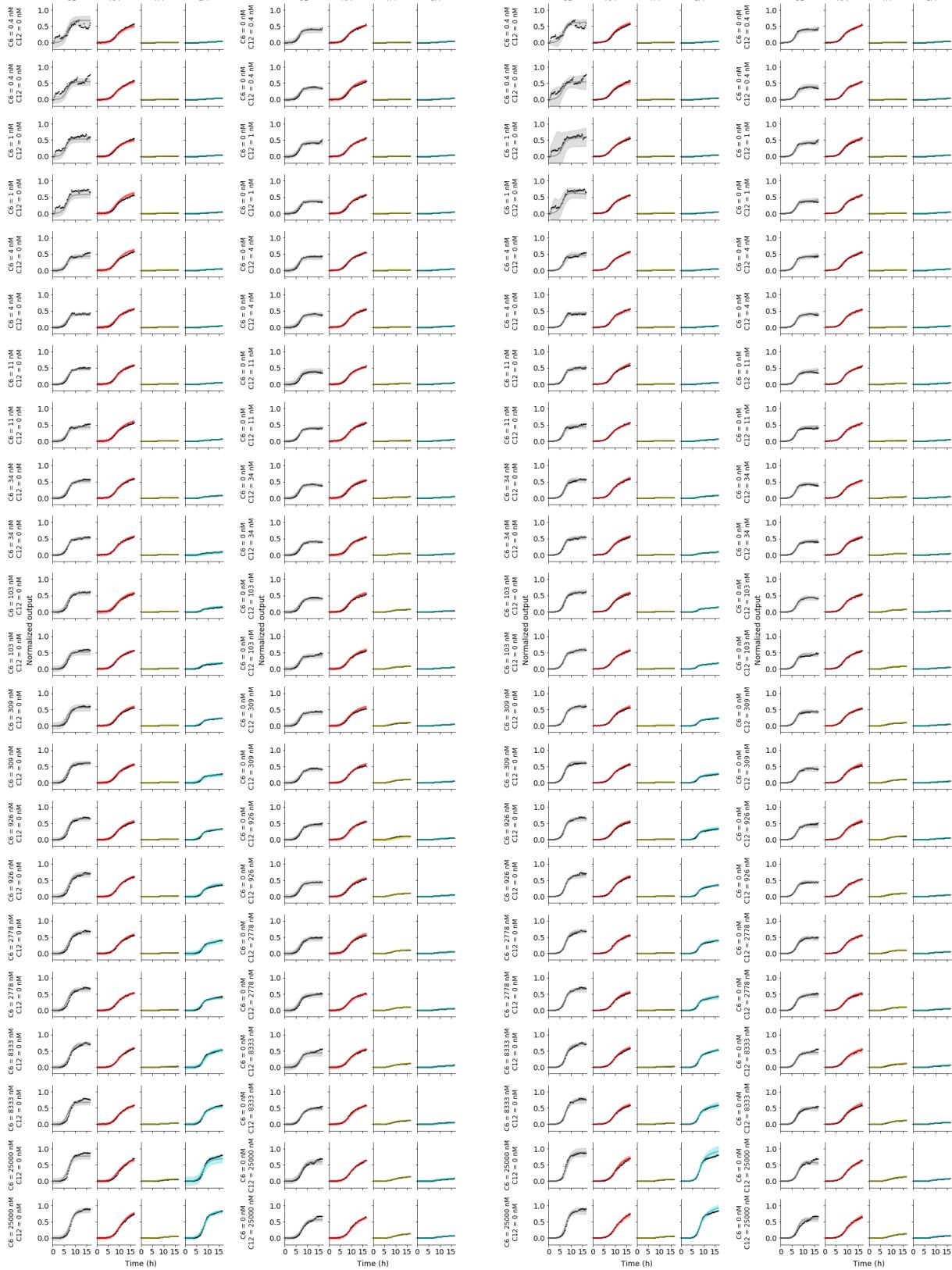


Figure 11. Posterior predictive distribution for 4-fold cross-validation experiment: R33-S32 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

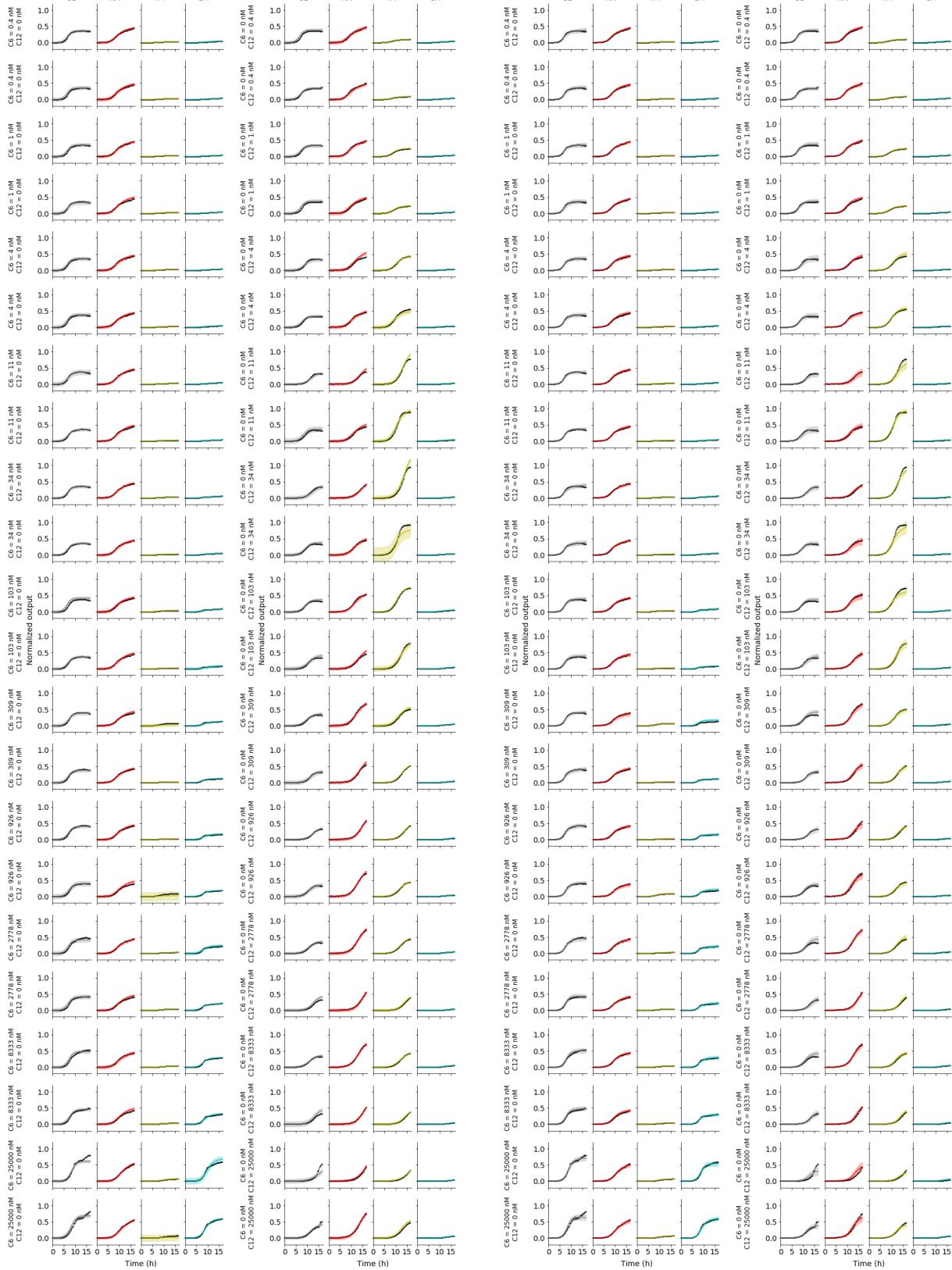


Figure 12. Posterior predictive distribution for 4-fold cross-validation experiment: R33-S34 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

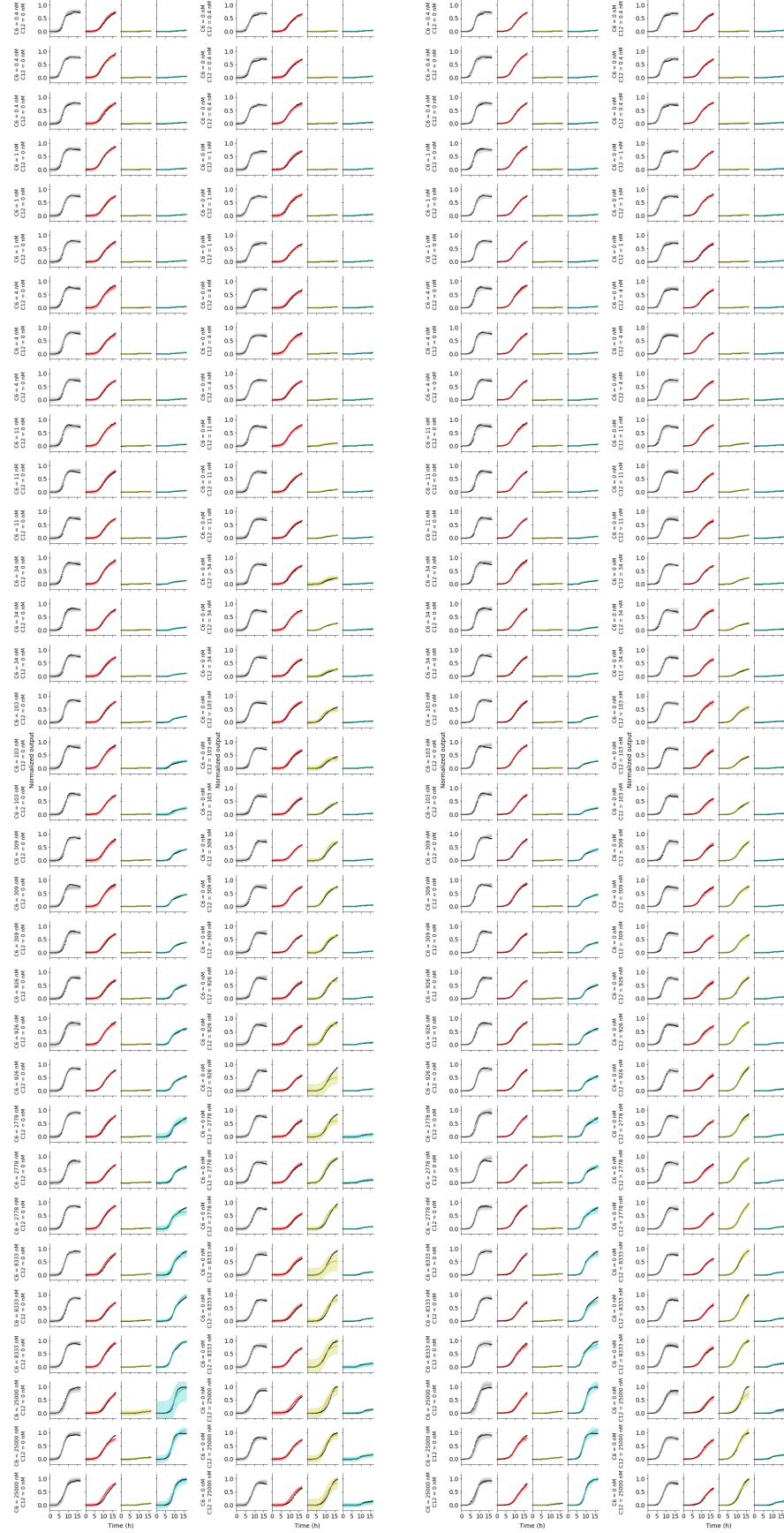


Figure 13. Posterior predictive distribution for 4-fold cross-validation experiment: R33-S175 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

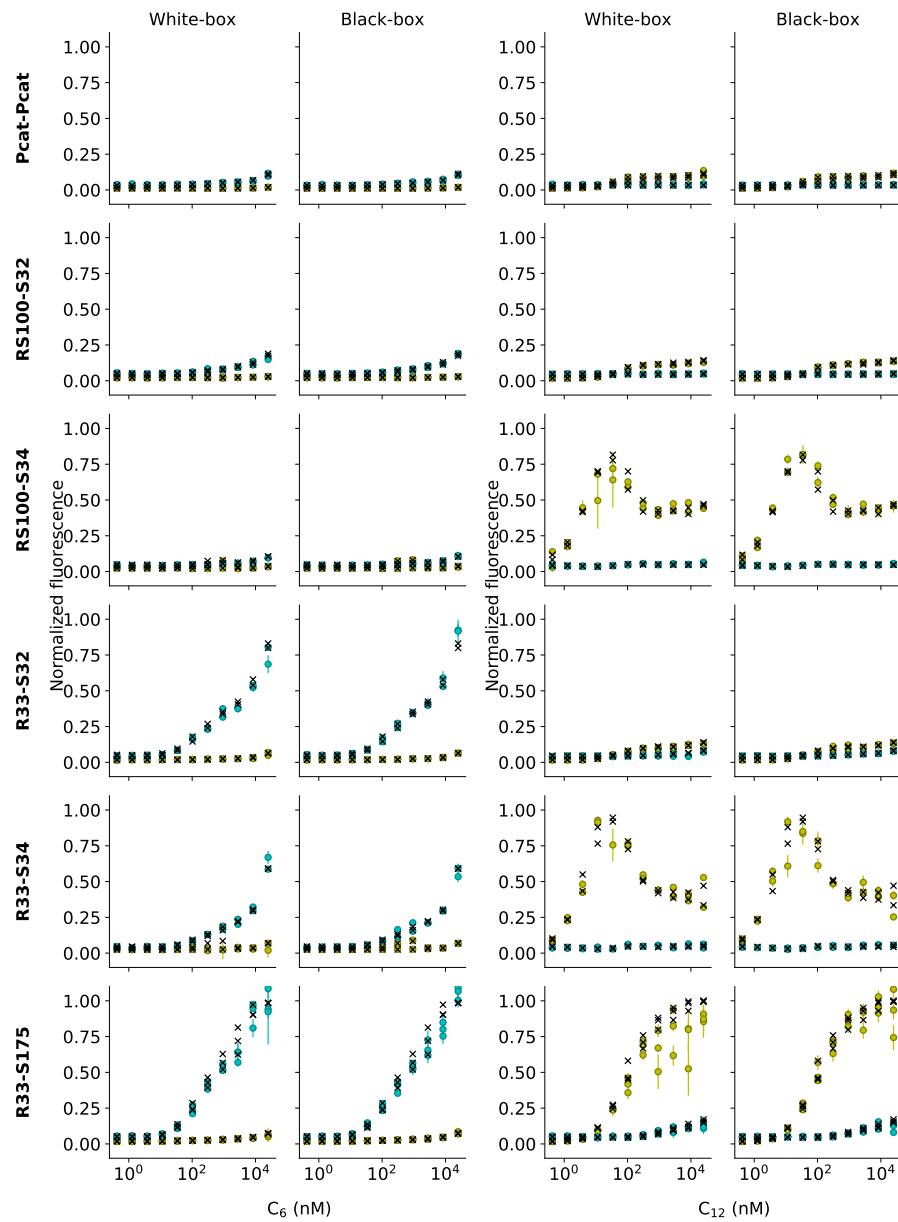


Figure 14. Treatment response plots

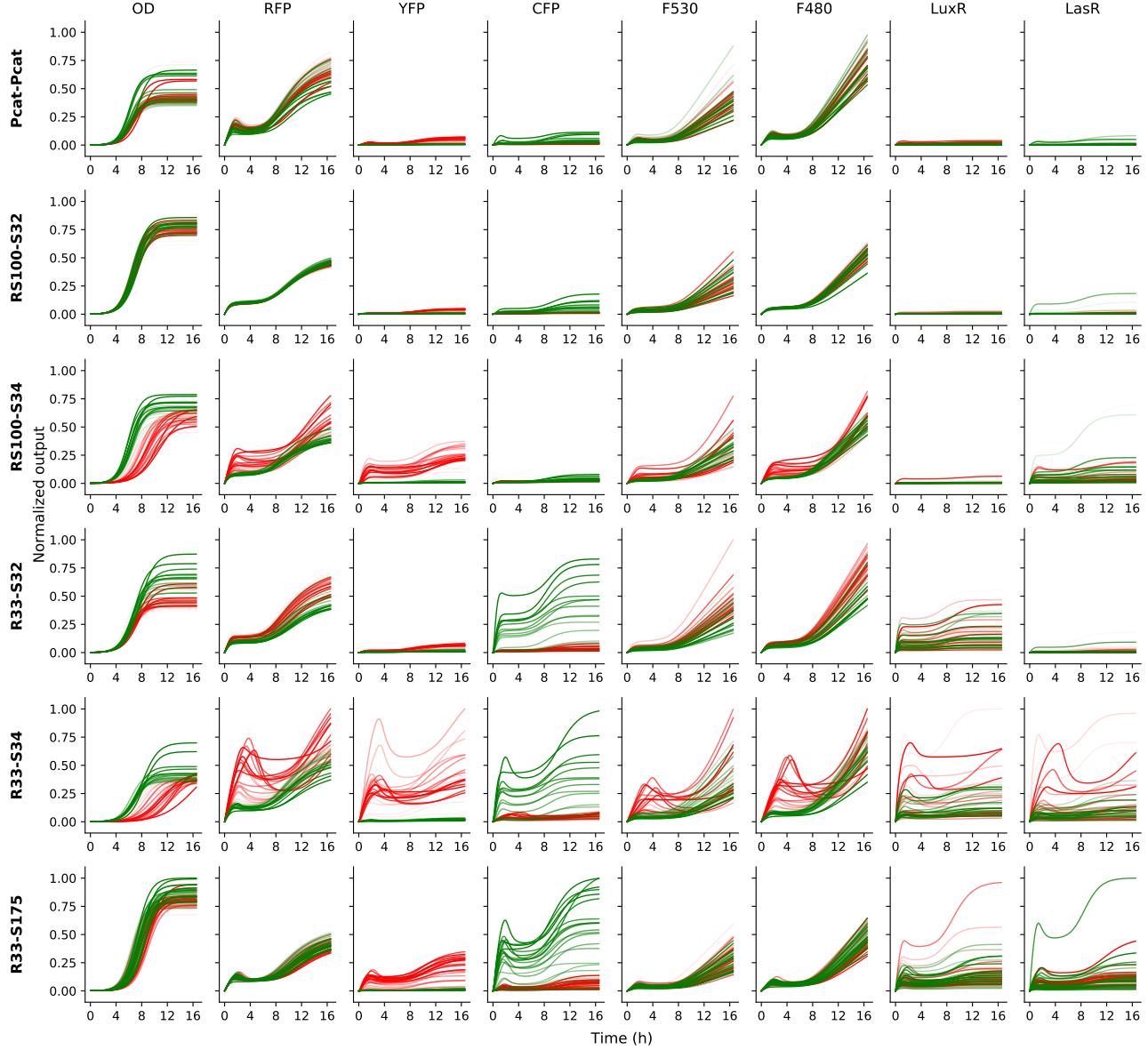


Figure 15. Predicted concentration dynamics of the hidden species (x) in the white-box model. In each panel, the traces are coloured according to whether they correspond to cultures treated with C_6 (green) or C_{12} (red).

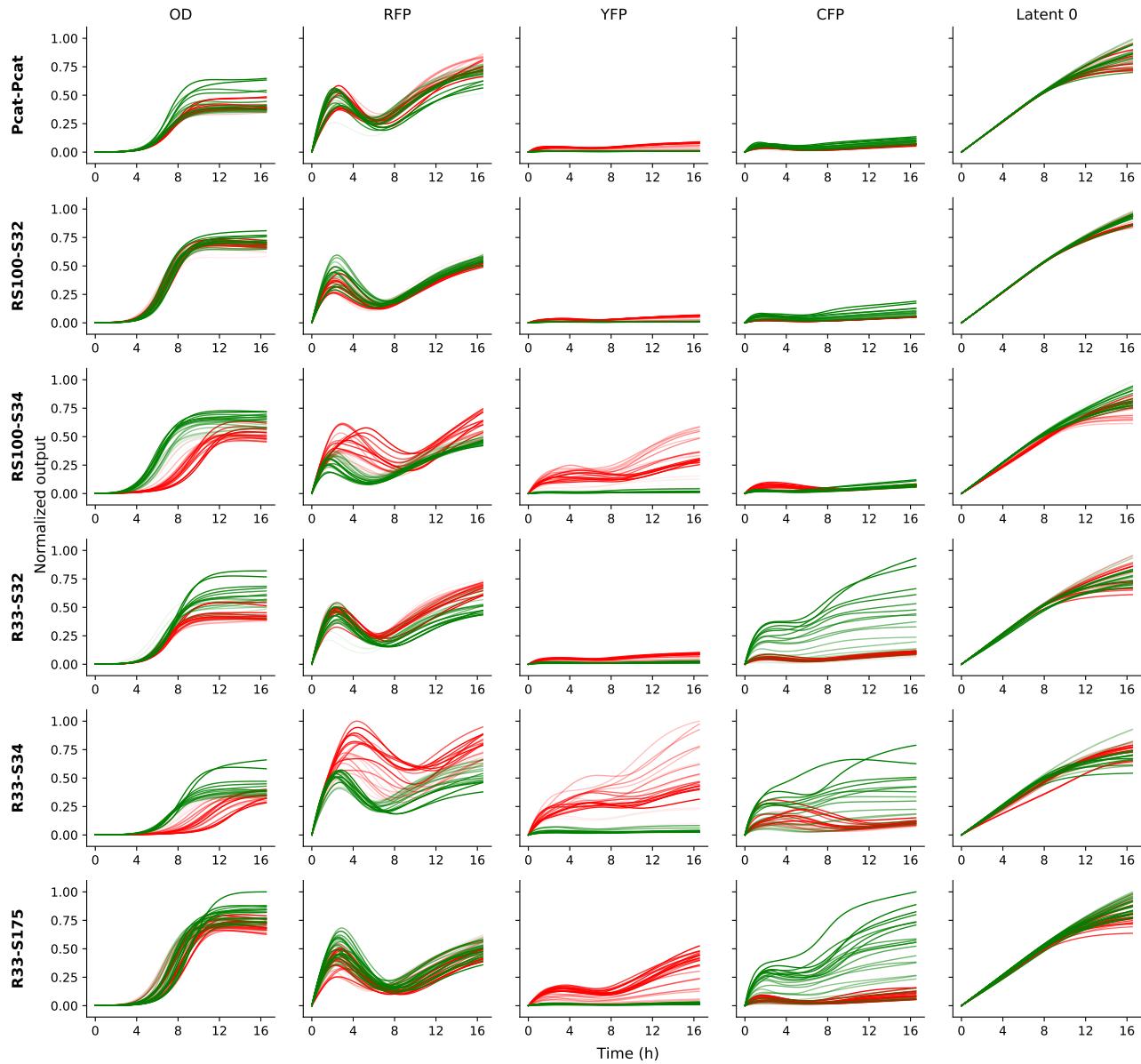


Figure 16. Predicted concentration dynamics of the hidden species (x) in the black-box model. In each panel, the traces are coloured according to whether they correspond to cultures treated with C_6 (green) or C_{12} (red).