
DATA-DRIVEN AND PHYSICS INFORMED MODELLING OF CHINESE HAMSTER OVARY CELL BIOREACTORS

A PREPRINT

Tianqi Cui^{1,4}, Tom Bertalan^{1,4}, Nelson Ndahiro¹, Pratik Khare¹,
Michael Betenbaugh¹, Costas Maranas², and Ioannis G. Kevrekidis^{1,3}

¹Department of Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, MD 21218, USA

²Department of Chemical Engineering, The Pennsylvania State University, University Park, PA 16802, USA

³Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218, USA

⁴These authors contributed equally.

ABSTRACT

Fed-batch culture is an established operation mode for the production of biologics using mammalian cell cultures. Quantitative modeling integrates both kinetics for some key reaction steps and optimization-driven metabolic flux allocation, using flux balance analysis; this is known to lead to certain mathematical inconsistencies. Here, we propose a physically-informed data-driven hybrid model (a “gray box”) to learn models of the dynamical evolution of Chinese Hamster Ovary (CHO) cell bioreactors from process data. The approach incorporates physical laws (e.g. mass balances) as well as kinetic expressions for metabolic fluxes. Machine learning (ML) is then used to (a) directly learn evolution equations (black-box modelling); (b) recover unknown physical parameters (“white-box” parameter fitting) or—importantly—(c) learn partially unknown kinetic expressions (gray-box modelling). We encode the convex optimization step of the overdetermined metabolic biophysical system as a differentiable, feed-forward layer into our architectures, connecting partial physical knowledge with data-driven machine learning.

1 Introduction

Chinese hamster ovary (CHO) cells are broadly used in biological and medical research, acting as the most common mammalian cell line used for the production of therapeutic proteins [1]. The advantage of using CHO cells is that the correct (i.e., mammalian-specific) glycosylation patterns are achieved for the protein therapeutics (e.g., therapeutic antibodies). Compared with conventional batch culture, fed-batch fermentation is more commonly used in this type of cell line, since it allows for easier control of the concentrations of certain nutrients that can affect the yield or productivity of the desired protein therapeutic molecule by ensuring the availability of precursor amino acids [2]. However, lack of a complete, clear, quantitative model of the metabolism becomes an obstacle to achieving accurate and precise system simulation and control.

In the past several decades, mathematical models that incorporate physical knowledge have been extensively applied in the analysis of cell metabolism [3, 4]. Metabolic Flux Analysis (MFA) leveraging stable carbon (i.e., ¹³C) labelled substrates techniques is the only technique that can provide information on internal fluxes [5, 6, 7]. Flux Balance Analysis (FBA), on the other hand provides a global inventory of carbon and energy resources throughout metabolism. By applying optimization principles, maximum theoretical yields for biomass formation or other products (e.g., metabolites or proteins) can be derived [8, 9]. Sometimes, for certain metabolic steps, detailed kinetic expressions are available that given the metabolite concentrations and enzyme levels can accurately estimate the flux through the metabolic reaction [10]. This requires the identification of the values of a number of enzymatic parameters. However, these expressions are usually available only for a subset of reactions, necessitating a hybrid modeling approach, where optimization is used to identify metabolic fluxes for the remainder of reactions that lack kinetic expressions. This gives rise to a system of ordinary differential equations (ODEs) determined by the stoichiometry of the reactions. In addition, given the fact that the metabolic reactions usually have relatively fast time constants (e.g. in the order

of milliseconds to seconds) compared with other cellular processes like growth and death of cells, the pseudo-steady-state assumption (PSSA) suggests that the accumulation rate of any and every intracellular metabolite can be usefully approximated as zero. For some reactions, (ir)reversibility can be posited based on thermodynamics considerations; for others, reaction rates can be estimated from chemical kinetics considerations [11].

Data-driven approaches are today increasingly employed for identification of complex system dynamics, including traditional regression methods as well as neural networks and their variants [12, 13, 14, 15, 16]. It is known, since the early 1990s, that neural networks embedded within numerical integrators can fruitfully approximate differential equations, and even learn corrections to approximate physical models, supplementing/enhancing them [13, 12, 17, 18, 14]. They can also be used to directly infer the evolution of the system variables when underlying physics are unclear [19, 20, 21, 22, 23]. Physics-Informed Neural Networks (PINNs) [24], Systems-Biology-Informed Neural Networks (SBINNs) [25, 26], and similar architectures [27], can and have been used to solve supervised learning tasks while respecting known laws of physics, system biology, et al [28]. Nevertheless, as we will discuss below, the ambiguous structure of metabolic models creates nontrivial technical difficulties in exploiting partially known physical information from experimental fed-batch culture metabolic data; and can drastically affect the training process for gray box neural networks trying to infer such models from experiments. Our goal in this paper is to elucidate the nature of these modeling ambiguities, demonstrating the ways in which they necessitate modifications of the architectures -and of the training- of traditional neural networks used for the identification task; and to implement networks capable of usefully identifying metabolic kinetics/parameters exploiting a synergy between physical modeling and scientific computation in neural network training.

2 Methods

2.1 Structure of the Biophysical Model

In a nutshell, the hybrid Chinese hamster ovary (CHO) bioreaction model we will use below (incorporating certain modifications (see appendix D and appendix E,) to the model presented in [10], which constitutes our starting point) describes a continuous-time dynamical system (the terms are defined in table 1):

$$\frac{d\mathbf{C}}{dt} = \mathbf{f}_{\text{eqn:ode}}(\mathbf{C}; \mathbf{v}(\mathbf{C}; \boldsymbol{\alpha})); \quad (1)$$

These evolution equations appear at first sight as simple ordinary differential equations (see appendix B for expressions of eq. (1)); yet, since evaluating the right-hand-side involves—as we will see—solving an optimization problem, we need another temporary label for the nature of the equations. Connecting with existing literature [29, 30, 9] we will here refer to these as **Dynamic Flux Balance Analysis** (DFBA) equations.

Here, $\mathbf{C} \in \mathbb{R}^K$ ($K = 14$) are variables tracked by experiments (which, though they might include concentrations of metabolites, cell densities, or other variables, we will simply refer to as “concentrations” for simplicity, see table 3); $\mathbf{v} \in \mathbb{R}^N$ ($N = E + I = 35$) are all fluxes (reaction rates, see appendix C for all reaction expressions) including I intracellular fluxes (which can be precomputed from the \mathbf{C}) $\mathbf{v}_I \in \mathbb{R}^I$ ($I = 14$) and E extracellular fluxes $\mathbf{v}_E \in \mathbb{R}^E$ ($E = 21$). Some of extracellular fluxes are assumed to be irreversible ($\mathbf{v}_{E,ir} \in \mathbb{R}^{E_{ir}}$ ($E_{ir} = 14$), $\mathbf{v}_{E,ir} \geq 0$), while others are assumed reversible ($\mathbf{v}_{E,r} \in \mathbb{R}^{E_r}$ ($E_r = 7$))); v is a function of \mathbf{C} and $\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^P$ ($P = 45$) are the kinetic parameters.

Notation	Variable	Dimension
\mathbf{C}	Variables tracked by experiments	$K = 14$
$\boldsymbol{\alpha}$	Kinetic parameters	$P = 45$
\mathbf{v}_I	Intracellular fluxes	$I = 14$
$\mathbf{v}_{E,r}$	Reversible (extracellular) fluxes	$E_r = 7$
$\mathbf{v}_{E,ir}$	Irreversible (extracellular) fluxes	$E_{ir} = 14$
\mathbf{v}_E	Extracellular fluxes	$E = E_{ir} + E_r = 21$
\mathbf{v}	All fluxes	$N = E + I = 35$
\mathbf{S}	Stoichiometric matrix	$M \times N = 24 \times 35$

Table 1: Notation and dimensions for all variables used (the fact that here, $K = 14, I = 14$ and $E_{ir} = 14$ is a coincidence).

Given C and α , the evaluation of $f_{\text{eqn:ode}}$ in eq. (1) is typically done in one of two very different ways. Both involve the following steps, but differ in the particular combination of objective/constraints and the optimization approach used to enforce them.

Given α , and an initial set of values C_0 for the concentrations, the time derivatives of the concentrations (e.g. RHS of Equation eq. (1)) can be computed via the following steps.

1. Compute preliminary updates of intracellular flux rates $\hat{v}_I \in \mathbb{R}^I$ ($I = 14$) according to the concentrations C and given formulas of kinetic equations

$$\hat{v}_I = f_{\text{kin}}(C; \alpha), \quad (2)$$

where $f_{\text{kin}} : \mathbb{R}^{K \times P} \mapsto \mathbb{R}^I$ (see appendix D for formulas of all kinetic expressions and appendix E for the changes of kinetic expressions we made based on the model in [10]).

2. The fluxes have to satisfy some constraints:

- Known kinetic expressions, i.e. Equation eq. (2).
- The pseudo steady state assumption, which requires

$$\mathbf{S} \cdot \mathbf{v} = 0, \quad (3)$$

involving the stoichiometric matrix $\mathbf{S} \in \mathbb{R}^{M \times N}$ ($M = 24$ is the number of metabolites at steady state, see appendix F for all entries of \mathbf{S}). If we split the columns of \mathbf{S} according to the I and E components (that is, $\mathbf{S}_I = \mathbf{S} \cdot \mathbf{B}_I$, $\mathbf{S}_E = \mathbf{S} \cdot \mathbf{B}_E$ where $\mathbf{B}_I \in \mathbb{R}^{N \times I}$ and $\mathbf{B}_E \in \mathbb{R}^{N \times E}$ are two indicator matrices showing the intracellular and extracellular indices of all reactions), we have an equivalent form of eq. (3),

$$\mathbf{S}_I \cdot \mathbf{v}_I + \mathbf{S}_E \cdot \mathbf{v}_E = 0$$

- Among all 21 extracellular fluxes \mathbf{v}_E , 14 of them are known to be irreversible, which requires

$$\mathbf{v}_{E,ir} = \mathbf{B}_{ir} \cdot \mathbf{v}_E \geq 0, \quad (4)$$

where $\mathbf{B}_{ir} \in \mathbb{R}^{E_{ir} \times E}$ is an indicator matrix containing the indices of irreversible fluxes among all extracellular ones.

Notice that the combination of Equations eq. (2) and eq. (3) consists of 38 independent linear equations, while the unknown variable \mathbf{v} is only 35-dimensional, leading to an overdetermined system. To address this issue, one can choose to satisfy some equations exactly, and others approximately (e.g. in a least squares sense). This leads to two substantially different approaches, the “kinetic-based” and the “stoichiometric-based”, for computing intracellular flux rates \mathbf{v}_I and extracellular \mathbf{v}_E . It is important to state that these two approaches will, in general, lead to substantially different dynamic evolution for the same initial conditions of a metabolic kinetic scheme.

- (a) The kinetic-based approach: we satisfy the kinetic equations eq. (2), and approximately satisfy the stoichiometric equations eq. (3), which leads to

$$\begin{cases} \mathbf{v}_I = \hat{\mathbf{v}}_I, \\ \mathbf{v}_E = \operatorname{argmin}_{\mathbf{v}_E} \|\mathbf{S}_I \cdot \hat{\mathbf{v}}_I + \mathbf{S}_E \cdot \mathbf{v}_E\|_2^2, \text{ s.t. } \mathbf{B}_{ir} \cdot \mathbf{v}_E \geq 0. \end{cases} \quad (5)$$

Here, we realize that the optimization problem is a linear least-squares problem with constraints (which implies it is actually a convex optimization problem). Moreover, if we ignored the constraints, we would be able to obtain an analytical solution for \mathbf{v}_E by computing the pseudo inverse of \mathbf{S}_E :

$$\mathbf{v}_E = -(\mathbf{S}_E)^+ \cdot \mathbf{S}_I \cdot \hat{\mathbf{v}}_I.$$

- (b) The stoichiometric-based approach [10]: we satisfy the stoichiometric equations exactly eq. (3), and then approximately satisfy the kinetic equations eq. (2), which leads to

$$(\mathbf{v}_I, \mathbf{v}_E) = \operatorname{argmin}_{\mathbf{v}_I, \mathbf{v}_E} \|\mathbf{v}_I - \hat{\mathbf{v}}_I\|_2^2, \text{ s.t. } \mathbf{S}_I \cdot \mathbf{v}_I + \mathbf{S}_E \cdot \mathbf{v}_E = 0, \mathbf{B}_{ir} \cdot \mathbf{v}_E \geq 0. \quad (6)$$

This is a least squares optimization problem with linear constraints. If we ignored the inequality constraints, we would obtain an analytical solution of $(\mathbf{v}_I, \mathbf{v}_E)$ by the Lagrange multiplier approach, see appendix G for details.

To help with the numerics of the two embedded optimization problems, we in fact rescale the supplied $\hat{\mathbf{v}}_I$ values: divide them by 1000 to adjust their numerical range to $\sim 1 - 10$ upon entering either optimization problem, and multiply the resulting fluxes \mathbf{v}_I and \mathbf{v}_E by the same factor before exiting. This does not change the solution, but improves the numerical conditioning.

After the optimization step we have

$$\mathbf{v} = \mathbf{B}_I \cdot \mathbf{v}_I + \mathbf{B}_E \cdot \mathbf{v}_E = f_{\text{optim}}^{(\ell)}(\hat{\mathbf{v}}_I), \quad (7)$$

where $\ell \in \{k, s\}$ indicates whether we are using the **kinetic** or **stoichiometric** approach to finding fluxes.

3. Compose eq. (1), eq. (7), and eq. (2) to create

$$\frac{d\mathbf{C}}{dt} = \mathbf{f}_{\text{eqn:ode}}(\mathbf{C}; \mathbf{f}_{\text{optim}}^{(\ell)}(\mathbf{f}_{\text{kin}}(\mathbf{C}; \boldsymbol{\alpha}))). \quad (8)$$

As the fluxes \mathbf{v} are the reaction rates for each of the $E + I$ reactions, this follows directly from the stoichiometry of these reactions.

Whether using the first or the second approach, the resulting set of equations can subsequently be integrated using an error-controlled integrator to obtain a full time series of all concentrations, for example,

$$\mathbf{C}(t = t_{n+1}) = \mathbf{f}_{\text{int}}(\mathbf{C}(t = t_n); \mathbf{f}_{\text{eqn:ode}}) = \mathbf{C}(t = t_n) + \int_{t_n}^{t_{n+1}} \mathbf{f}_{\text{eqn:ode}}(\mathbf{C}; \mathbf{v}(\mathbf{C}; \boldsymbol{\alpha})) dt, \quad (9)$$

where $\{t_i : i = 0, 1, 2, \dots\}$ is the set of equal-spaced timestamps. It is important however to note that rate discontinuities potentially can (and actually do) arise at time instances when different constraints become active (see fig. 3). Note also that typical operating protocols of bioreactors often call for the addition of species (e.g. nutrients) at particular time instances, thus leading to temporal discontinuities in the system states. We will illustrate both these types of discontinuities below in section 3.1.

Several important contributions on which this paper is based were established in previous work, beginning with applications to *E. coli* [9] and then proceeding to the more recent mammalian biomanufacturing targets [10]. Beyond the constraints on our inner (optimization) problem that we showed above, additional constraints were imposed in [9] on their outer (time-integration) problem, such as non-negative metabolites and limits on the rate-of-change of fluxes. In their dynamic (resp. static) optimization approach (DOA, resp. SOA) they determined fluxes over an entire trajectory (resp. one trajectory segment, with constant fluxes). Our simulations can be thought of as a form SOA, with the segment being a single integration step (as also in [10]).

Before we start, a note on the computation of model gradients: many accurate integrators require the system Jacobian as well as sensitivities w.r.t. parameters. This is also important for the integration of differential-algebraic systems of equations (differential equations with equality constraints). Furthermore, these gradients (w.r.t. state variables and/or parameters) are crucial in identification tasks: training neural networks to approximate the system equations and/or estimate their parameters from data. As we described above, our evolution equations are not simple explicit ordinary differential equations, but rather, their right-hand side arises as the result of solving an optimization problem, depending on the current state. This renders the accurate evaluation of these ODEs (as well as their sensitivity and variational computations) less straightforward than the explicit right-hand-side case.

2.2 Black-box Model

Our black-box model is a multi-layer perceptron (MLP) embedded within a numerical integrator scheme (e.g. the forward-Euler scheme or the Runge-Kutta template), where the MLP ($\text{NN}_b(\cdot; \boldsymbol{\theta})$) is used to learn the right-hand-side (RHS) of the ODE:

$$\tilde{\mathbf{C}}(t = t_{n+1}) = \mathbf{f}_{\text{int}}(\mathbf{C}(t = t_n); \text{NN}_b) = \mathbf{C}(t = t_n) + \int_{t_n}^{t_{n+1}} \text{NN}_b(\mathbf{C}; \boldsymbol{\theta}) dt. \quad (10)$$

The details of generating the datasets can be found in section 3.1. Note that here the right-hand-side depends only on the system state; it is also possible to make the Neural Network eq. (10) dependent on physical input parameters (such as feeding conditions or basal gene expression rates), by including these parameters as additional inputs to the NN function. This will be important if, at a later stage, one wishes to optimize operating conditions towards some additional global objective (e.g. maximal biomass production). This possibility has been demonstrated in older work [17]; it will not be repeated here.

2.3 White-box and Gray-box Models

2.3.1 Model Structures

In contrast with the black-box model which is purely data-driven, white-box and gray-box models benefit from existing physical knowledge, leaving only the unknown parts of the model trainable. In this paper, these two models have structure similar to that of what we deem the ground-truth biophysical model (see fig. 1), with changes limited to the computation of preliminary intracellular fluxes $\dot{\mathbf{v}}_I$. While the white-box model assumes that some of the kinetic parameters $\boldsymbol{\alpha}$ are unknown or need calibration, the gray-box model suggests that part of the kinetic expressions \mathbf{f}_{kin}

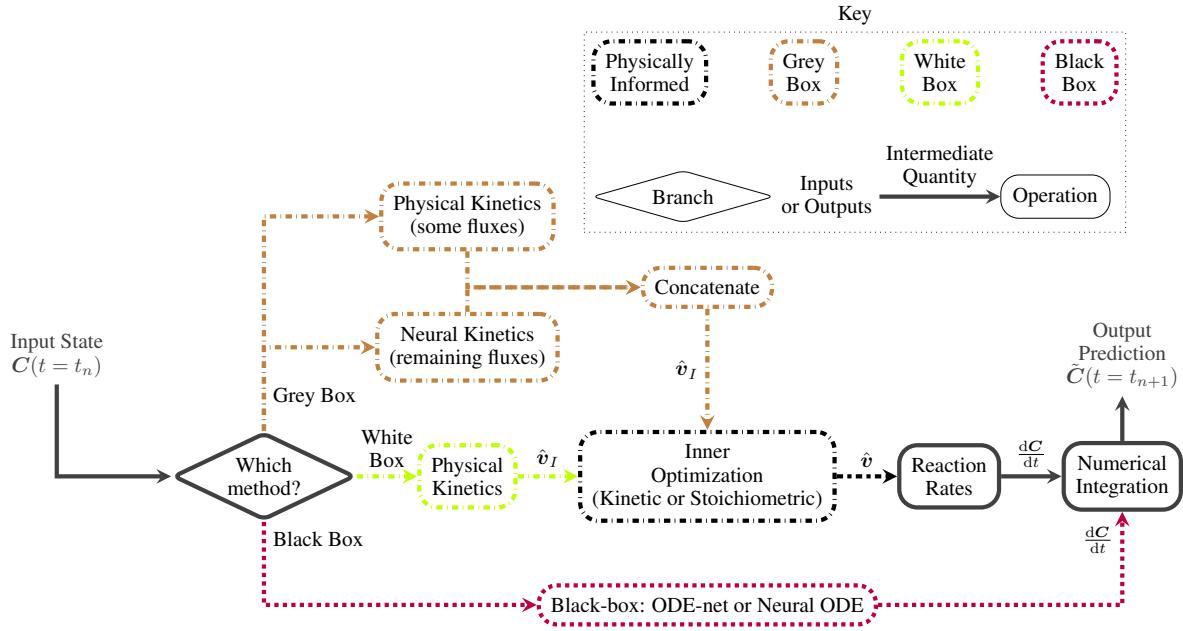


Figure 1: **White, gray, and black inner architectures.** Operations are boxed, data or predictions are unboxed, and notable named intermediates are labeled on edges. Color and pattern are used to distinguish between model pathways that are distinct between the gray-, white-, and black-box approaches, or common to all three.

have no known functional form and therefore replaces them with neural network approximations. It is natural to also construct a mixed version of the hybrid model that contains both unknown (“white-box”) kinetic parameters and unknown (“black-box”) kinetic expressions, resulting in an overall gray-box model.

Though the white-box model superficially resembles typical parameter-fitting problems, due to the presence of the inner optimization step in its evaluation, traditional fitting approaches like general linear least-squares cannot be well-adapted into our framework. Instead, a gradient-based fitting approach will be employed using a differentiable convex optimization layer as described in section 2.3.2. We remind the reader here that, since the original biophysical model includes two different approaches for computing fluxes (see eq. (7)), we also make our white- or gray-box frameworks in two versions: that is, we use kinetic-version models to learn on the dataset generated from the kinetic approach, and stoichiometric-version models on the dataset that came from the stoichiometric approach.

2.3.2 Computation of Gradients in Convex Optimization

For this section in particular, we need to define some terms.

The *model* refers to the differentiable program used to make predictions: this includes RHS evaluations (white-, gray-, or black-box), perhaps necessitating an embedded convex optimization program eq. (5) or eq. (6) (**ECOP**); as well as the use of these RHS in numerical integration steps.

The *inputs* for this ECOP include both

$$\begin{array}{ll} \text{constants } \mathbf{S}_I, \mathbf{S}_E, \text{ and } \mathbf{B}_{ir}; & \text{and} \\ \text{outputs from upstream modules } & \hat{\mathbf{v}}_I \quad (\text{function evaluations}). \end{array} \quad (11)$$

All of these will be considered constant for the purpose of solving the ECOP for each call to the RHS.

Parameters here refers to those quantities which could be modified by our outer training loop, including both kinetic parameters α of the kinetic equations eq. (2) (when we perform white box parameter estimation or gray box parameter estimation); and neural network parameters θ , i.e. trainable weights and biases (when we train gray box networks to

recover unknown functional dependencies):

$$\alpha \quad \text{and} \quad \theta. \quad (12)$$

We will divide these into trainable and untrainable (fixed) parameters depending on the particular experiment.

The *outputs* of the ECOP are the reported converged values of the *variables* the problem solves for, including both

$$\begin{aligned} & \text{the fluxes } \mathbf{v}_E \text{ and } \mathbf{v}_I; \quad \text{and, possibly} \\ & \text{auxiliary variables (described below)} \quad \mathbf{r}. \end{aligned} \quad (13)$$

For the purposes of training, we would like our loss function (see eq. (16) described for particular experiments in later sections) to be differentiable with respect to all of the trainable parameters eq. (12). This requires that *the model predictions* be differentiable, and therefore for each step in the model to be differentiable, including the ECOP, with respect to the same.

To enable this in a gradient-based computing framework such as PyTorch, we turn to the package `cvxpylayers` which was developed with such problems in mind [31]. This package itself uses `cvxpy` (a Python-embedded modeling language for convex optimization problems) [32, 33, 34] and `diffcp` (a Python package for computing the derivative of a cone program, which is a special case of convex programming) [35, 36, 37].

In order for these packages to correctly evaluate the gradient of the outputs eq. (13) of the ECOP with respect to all of its inputs eq. (11), certain structural characteristics must hold. Specifically, the problem needs to be rewritten conforming with the rules of Disciplined Convex Programming (DCP) and of Disciplined Parametrized Programming (DPP). DCP is a system for constructing convex programs that combines common convex functions (e.g. x^2 , $|x|$) with composition and combination rules (e.g. $f \circ g$ is convex if both f and g are convex; nonnegative linear combination of convex functions is still convex). If these rules are followed, the library can automatically determine whether the full problem is indeed convex.

DPP is a subset of DCP, which further requires that all expressions of the ECOP are affine with respect to the ECOP inputs eq. (11). It has been proved [34] that a DPP-supportable convex program can be invertibly transformed into a cone program (and its derivative information can be obtained from `diffcp`). Therefore, DPP is mainly used in input-dependent convex programming, which allows the entire program to be differentiable without actually unrolling and back-propagating through the optimization loop.

Because DCP requires that the expressions in the ECOP be affine w.r.t. the problem inputs eq. (11), the product of two inputs is not an acceptable expression. This means e.g. that $(\mathbf{S}_I \cdot \hat{\mathbf{v}}_I + \mathbf{S}_E \cdot \mathbf{v}_E)^T \cdot (\mathbf{S}_I \cdot \hat{\mathbf{v}}_I + \mathbf{S}_E \cdot \mathbf{v}_E)$ in the objective function of eq. (5) has to be reformulated. This is resolved by the addition of another variable \mathbf{r} in eq. (14) and eq. (15), and then equality constraints on this additional variable, such that the newly defined problems are equivalent to the old. Further, in the kinetic case, to avoid the direct input product $\mathbf{S}_I \cdot \hat{\mathbf{v}}_I$, we need to include \mathbf{v}_I as an optimization variable, but then upgrade what was a pre-optimization expression $\mathbf{v}_I = \hat{\mathbf{v}}_I$ from eq. (5) to an actual equality constraint in eq. (14).

In summary, for the kinetic approach, we rewrite eq. (5) as

$$\begin{aligned} & \min_{\mathbf{r}, \mathbf{v}_I, \mathbf{v}_E} \|\mathbf{r}\|_2^2 \\ & \text{s.t. } \mathbf{B}_{ir} \cdot \mathbf{v}_E \geq 0 \\ & \quad \mathbf{r} = \mathbf{S}_I \cdot \hat{\mathbf{v}}_I + \mathbf{S}_E \cdot \mathbf{v}_E \\ & \quad \mathbf{v}_I - \hat{\mathbf{v}}_I = 0 \end{aligned} \quad (14)$$

and for the stoichiometric approach, we rewrite eq. (6) as

$$\begin{aligned} & \min_{\mathbf{r}, \mathbf{v}_I, \mathbf{v}_E} \|\mathbf{r}\|_2^2 \\ & \text{s.t. } \mathbf{B}_{ir} \cdot \mathbf{v}_E \geq 0 \\ & \quad \mathbf{r} = \mathbf{v}_I - \hat{\mathbf{v}}_I \\ & \quad \mathbf{S}_I \cdot \hat{\mathbf{v}}_I + \mathbf{S}_E \cdot \mathbf{v}_E = 0 \end{aligned} \quad (15)$$

with inputs eq. (11).

With these changes, the two problems are DPP-compliant; so, we are able to evaluate derivatives of the problem outputs eq. (13) (in particular, the argmins \mathbf{v}_I and \mathbf{v}_E) with respect to the problem inputs eq. (11) and also evaluate vector-Jacobian products as needed in a larger PyTorch back propagation to eventually get loss gradients w.r.t. parameters eq. (12).

2.4 Auto-regressive Loss

For supervised learning of the dynamics underlying time series data, one approach is to use the ground truth prediction/output from a prior time step as the input for the current time step, which leads to the teacher-forcing method (also known as professor-forcing in [38]). Alternatively, we could use the model prediction from the prior time step as input, which is called “autoregressive training”. In fact, if contiguous data trajectories are divided into episodes of M steps each, and M reduced to 2, we see that the first is in fact a special case of the second. So, in general, we use an autoregressive model structure (however, see also eq. (10)), which means the forward pass of the model can be written as

$$\tilde{\mathbf{C}}(t = t_{i+1}) = \text{Model}(\tilde{\mathbf{C}}(t = t_i)), \tilde{\mathbf{C}}(t = t_0) = \mathbf{C}(t = t_0), \quad (16)$$

where Model can represent the integration of the black-, white- or gray-box RHS. The mean-squared loss (MSE) between the two time series can be therefore computed as

$$\text{MSE}(\{\tilde{\mathbf{C}}(t = t_i)\}, \{\mathbf{C}(t = t_i)\}) = \frac{1}{KL} \sum_{j=1}^L \|\tilde{\mathbf{C}}(t = t_j) - \mathbf{C}(t = t_j)\|_2^2, \quad (17)$$

where $L + 1$ is the length of the dataset $\{\mathbf{C}(t = t_i) : i = 0, 1, 2, \dots, L\}$ and $K = 14$ is the dimension the state vector as we have shown in table 1.

3 Results

In this section, we will describe each of the several computational experiments tabulated in table 2 which we performed in this paper. We first begin by describing the data generation procedure that was used for each of the parameter-identification and neural-network training experiments that follow. We include an analysis of the impact of the constraints in the inner optimization problem, considering events when constraints switch to (resp. from) active (resp. inactive). We then begin our actual training experiments with a black-box example. All of our training experiments include both kinetic and stoichiometric variants. Subsequently, we perform white-box identification, in both two-free-parameter and five-free-parameter variants. Finally, we will perform a mixture of these two tasks with gray-box modeling: First we will use a neural network to replace one of the kinetic expressions in f_{kin} ; then we will repeat this, also allowing one of the physical parameters α to the trainable.

Experiments	Section
Data Generation	§3.1
Black-box	§3.2
White-box (2 Parameters Unknown)	§3.3.1
White-box (5 Parameters Unknown)	§3.3.2
Gray-box (1 Expression Unknown)	§3.4.1
Gray-box (1 Expression Unknown + 1 Parameter Unknown)	§3.4.2

Table 2: Summaries of computational experiments in this paper.

3.1 Data Generation

We begin by simulating short trajectories for a variety of initial conditions, and collecting these flows as a dataset for a single set of parameter values. We then implement the Neural Network model in PyTorch exactly as described in section 2.1 and trained to match these flows.

The dataset consists of N_{run} transients of the full model eq. (1), or equivalently, eq. (8), from initial conditions (ICs) taken as Gaussian perturbations around means; the means themselves are sampled uniformly (in time) at random along a central nominal trajectory (NT). The per-variable standard deviations are proportional to the extent of variation of that variable in the NT. That is, the sample of ICs is given by

$$\left\{ \mathbf{C}^{(i)}(t = 0) = \begin{bmatrix} C_1^{(i)}(t = 0) & \sim & \mathcal{N}(\bar{C}_1(t = t_i), \sigma_1) \\ \vdots & & \\ C_K^{(i)}(t = 0) & \sim & \mathcal{N}(\bar{C}_K(t = t_i), \sigma_K) \end{bmatrix} \middle| i = 1, 2, \dots, N_{\text{run}} \right\}, \quad (18)$$

where the nominal trajectory $\bar{\mathbf{C}}(t) = (\bar{C}_1(t), \bar{C}_2(t), \dots, \bar{C}_K(t))^T$ starts from a particular set of initial conditions that were measured during a laboratory experiment. The feeding events were implemented as state discontinuities. This

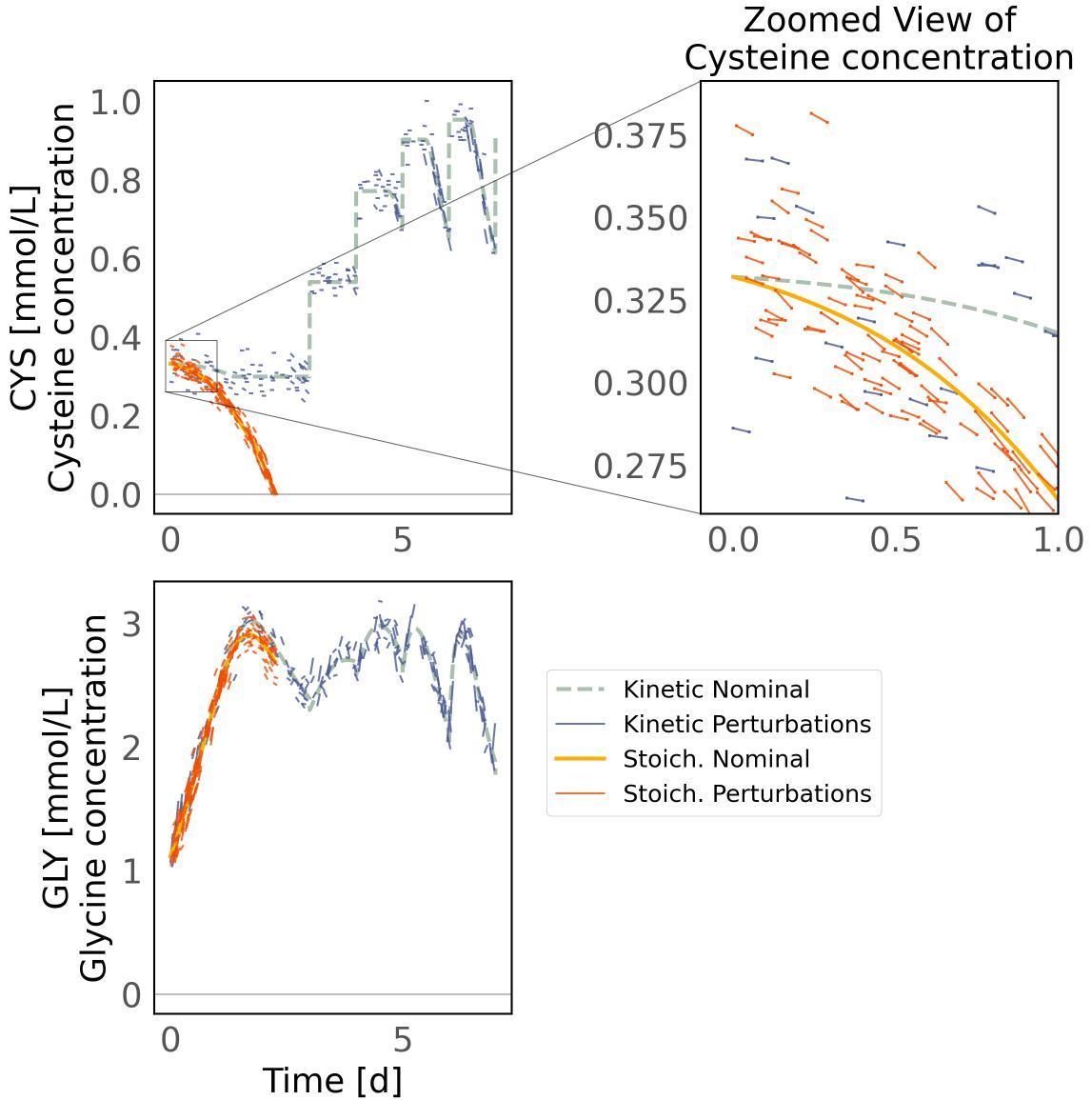


Figure 2: Nominal trajectories (kinetic and stoichiometric) overlaying sampled short-time flows from perturbed initial conditions. Trajectories for only a few of the K variables are shown. Trajectories for all variables are shown in fig. 15. Color is used to distinguish between different curves described in the legend.

nominal trajectory (in both its “kinetic” and its “stoichiometric” integrations) appears in fig. 2; the “stoichiometric” NT stops just before the state goes negative (and thus before any feeding events have occurred).

Each initial condition from eq. (18) is then accurately simulated (with an order 8(5, 3) explicit Runge-Kutta method [39] with absolute tolerance 10^{-8} and relative tolerance 10^{-7}) to a time horizon generally significantly shorter than the entire nominal trajectory, (circular dots in fig. 4 and fig. 5) giving us data in the form of several trajectory “windows”, constituting one “episode”.

3.1.1 Detecting transitions in constraint activity during simulation

The inner optimization described in section 2.1 includes bounds on some of the fluxes computed (lower bounds indicating irreversible reactions). Depending on the current state of the simulated variables C , the optimal *unconstrained*

fluxes may not lie inside the bounded domain. The *constrained* optimum will then instead lie on constraint boundaries (or even possibly intersections of them).

At the onset—or the end—of occurrence of such events, the trajectory of v may/will develop sharp corners. To explore the impact of this phenomenon on the system dynamics we report, at each timestep of a simulation, which flux bounds are active and which are not. For an event-driven version of such simulations we refer the reader to the package in [30], in which several failure modes are considered (beyond the ones arising in our work), including both an infeasible inner optimization problem, and a problem with multiple solutions (leading to a set-valued differential equation). In general, events can be either “time events” or “state events”; the first require only accurately stopping the integration at a particular time. State events, on the other hand, occur when some condition(s) of the continuous state become satisfied. In some cases, this can be detected by locating zeros of some interpolating polynomial(s) [30, 40]; further difficulty arises when the event condition cannot be described by the root of a continuous function (e.g., the case of fig. 3, in which a Boolean quantity changes at the event). We intend to explore the proper analysis of such event detection in future work (as well as the integration between events, possibly modifying the RHS evaluation between each pair of events to satisfy the active bounds by construction). We employ instead a less sophisticated visualization-based method, shown in fig. 3: We depict, in fig. 3, such changes in constraint activity status alongside the first (3c, 3a) and second (3d, 3b) time derivatives of two key simulated variables in a temporally aligned fashion. This is shown here along a short run of the stoichiometric model; a constraint turning active is marked by a short green tick, and its turning inactive by a brief red tick). Notice the jumps in the second derivative, and the sharp corners in the first derivative of the concentration evolution.

During this run we observe that three fluxes (7, 32 and 34) had encounters with their corresponding lower bounds. More specifically, around $t^* \approx 0.5$, when the bound for reaction 34 becomes *persistently inactive* (thus the corresponding flux moves well away from zero), we see sharp downward discontinuities in the second time derivative of the evolutions of both cysteine and glycine. Note that reaction 34 is the (irreversible) breakdown of NADH (see full stoichiometry matrix in appendix F, or the relevant parts of the reaction network diagram in [10]); and that, at this time, its flux continuously changes from 0 to positive. We therefore expect, dv_{34}/dt may experience a discontinuity at t^* . Furthermore, one of the (reversible) reactions that makes GLY takes NADH as an input. If that reaction rate is positive at that given moment, one of its inputs suddenly becomes less available. Therefore, because $d\text{GLY}/dt \sim -v_{34}$, and dv_{34}/dt is discontinuous, we can expect that $d(d\text{GLY}/dt)/dt$ will also be discontinuous, and that is clearly visible in fig. 3b.

Such a rationalization can be repeated for CYS, which also relies on NADH as an input, and which also experiences a discontinuity in its second derivative (fig. 3d). Note that CYS has a much larger discontinuity associated with the activation of the lower bound on flux 32. Note also that GLY has a second discontinuity occurring later (around $t \approx 1.4$); this is related to flux 7, which however involves different pathways.

Data-driven and Physics Informed Modelling of Chinese Hamster Ovary Cell Bioreactors

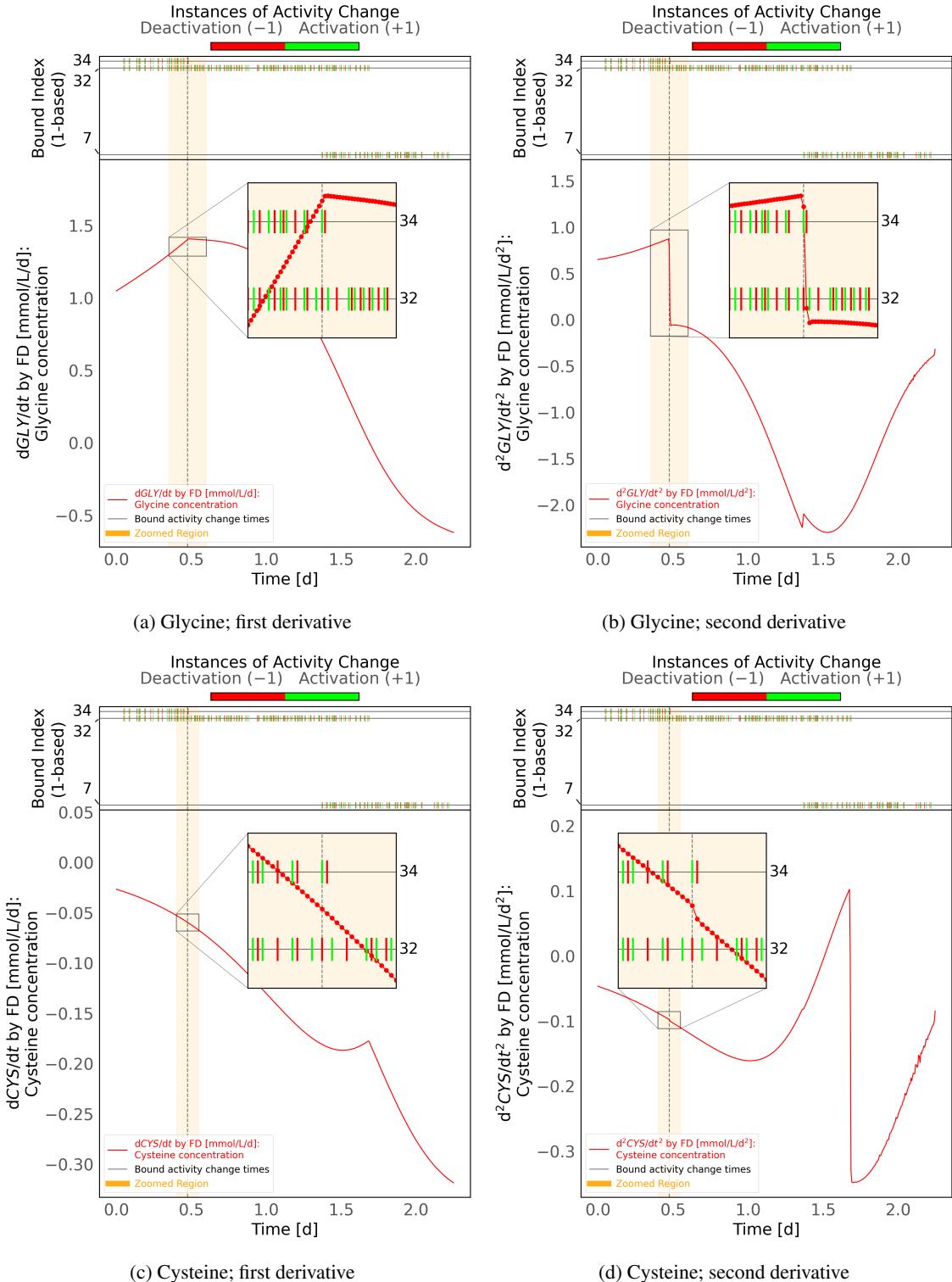


Figure 3: **Activity of bound constraints along a sample run for the stoichiometric case**, Observe the discontinuities arising in the second derivative (3b and 3d) of the concentration evolution. Time derivatives for plotting were estimated by local forward finite differences (FD). Color is used to distinguish between different curves described in the legend.

3.2 Black-Box Neural Network Identification

To demonstrate system identification with no assumed prior knowledge of the system mechanisms, we performed black-box RHS learning, in which we represent the entire system of ODEs as an end-to-end neural network.

Since there are two approaches in evaluating eq. (8) as mentioned in section 2.1, we also performed two experiments: one of them used data generated from the ground-truth kinetic system (sampling every $\Delta t = 0.1$ hours over a $t_{\max} = 1.2$ -hour horizon, producing 13 steps for each of the 768 data trajectories), and the other used data generated from the ground-truth stoichiometric one (with $\Delta t = 0.1$, $t_{\max} = 1.2$, 13 steps, and 768 data trajectories).

In both cases, the black-box ODE was trained by taking steps of fixed size 0.01 between the data samples with a Runge-Kutta 4 integrator (the black box identification “does not know” about discontinuities in the model - it smoothly interpolates between data points in time).

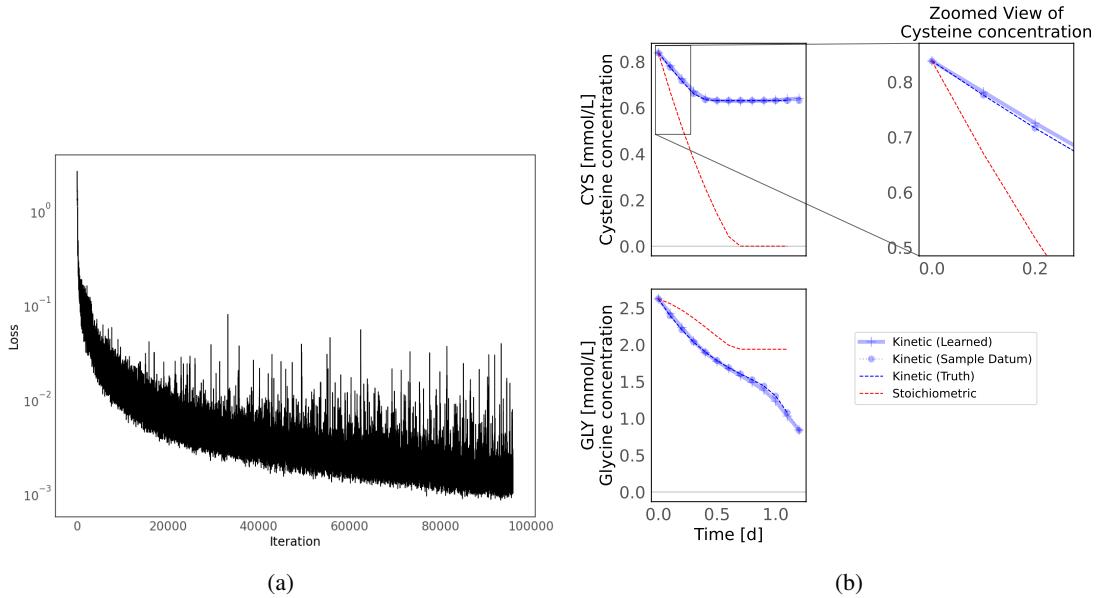


Figure 4: Black-box training results (kinetic). See also fig. 16 for more detailed results. Color is used to distinguish between different curves described in the legend.

As can be seen in fig. 4 and fig. 5, our black-box neural ODE was able to fit the data trajectories quite tightly. This validates the underlying approach, and suggests that such ODEs with inner optimization steps can be successfully approximated as (possibly slightly “smoothed”) closed-form functions.

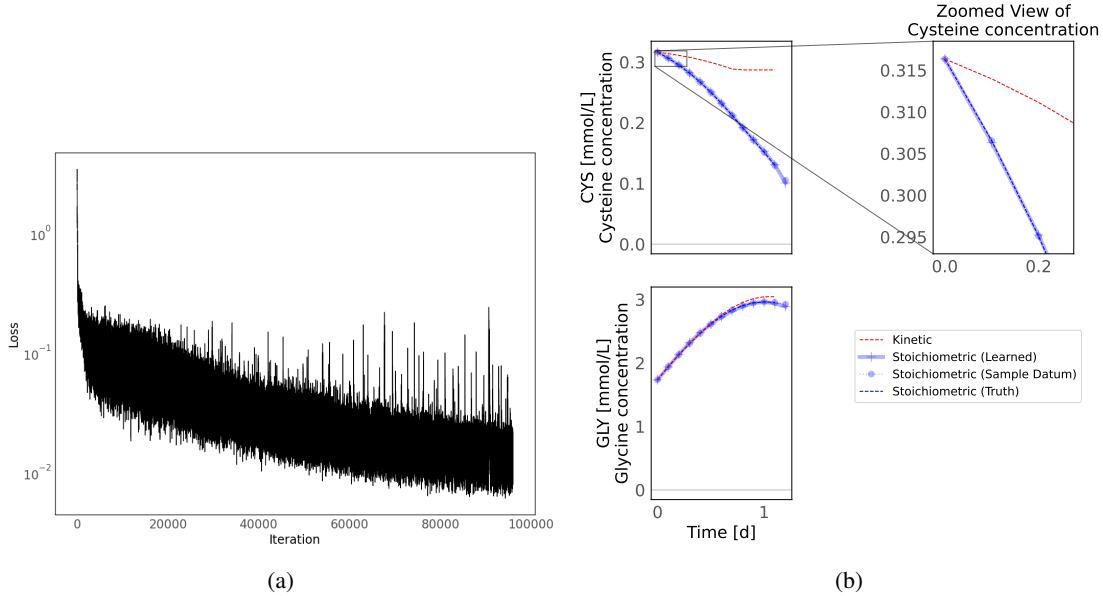


Figure 5: Black-box training results (stoichiometric). See also fig. 17 for more detailed results. Color is used to distinguish between different curves described in the legend.

3.3 White-Box Neural Network: Parameter Estimation

To demonstrate the full-structure physical-parameter estimation setting that we term “white-box” learning, we tried to recover (a) two or (b) five of the nominal parameter values. Specifically, we performed simulations at the nominal parameter values, collected the transient data, considered forty three (resp. forty) of them known, and then used a gradient-based training method to estimate the values of the remaining two (resp. five) from the data. Our initial guess (a perturbation of the truth) is marked in fig. 7. For all four of these numerical experiments, the dataset consisted of 10 short single-Euler-step “trajectories” each 0.05 hours in length (Runge-Kutta integration gave comparable results, not shown); the network ansatz was also Euler with a step size of 0.05. Training was 4000 epochs of RMSprop with 1 batch per epoch.

This demonstrates the use of the algorithms in [31] to carry out differentiation through the inner optimization problem of evaluating the equation right-hand-side, discussed in section 2.1, enabling gradient-based learning for this experiment, and serving as an initial validation of the algorithms before the gray-box methods of section 3.4 that follow below.

3.3.1 Known Model, Two Unknown Parameters

In our first such white-box learning experiment, we trained with only two unknown parameters from table 4; we find (fig. 6) that we can recover the two parameter values reasonably well. A motivation for this initial experiment is to help visualize the gradient landscape in fig. 7. We see in fig. 8 that the induced gradient dynamics of the learning problem are highly stiff, making adaptive training methods such as Adam [41] an absolute necessity. The training exhibits a two-staged descent, consisting of (a) first, a fast approach to a deep trough in the parameter space, and then (b) a slower motion within the trough, with some oscillations induced by the finite learning step size. Furthermore, in the Stoichiometric (Type 2) case (fig. 7b and fig. 8b), we observe that the final gradient is so shallow that even Adam takes prohibitively long to move any appreciable distance within the loss trough. Note that, although the true α values indeed mark a minimum for the optimization problem posed, as seen in fig. 7b, this minimum is extremely shallow, making the discovery of the true value for α_2 imperfect - for all practical purposes, the entire “bottom of the trough” leads to a good fit. This is an instance of what is termed “model sloppiness” [42, 43]; along the bottom of the trough the loss function posed is not strongly sensitive to the parameters, leading to parameter nonidentifiability.

Data-driven and Physics Informed Modelling of Chinese Hamster Ovary Cell Bioreactors

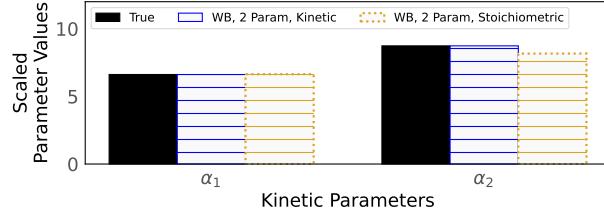
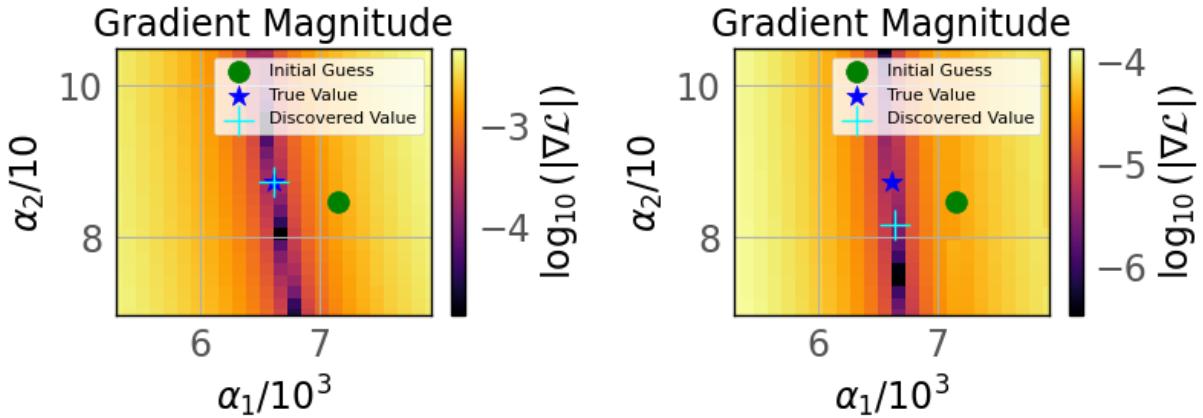
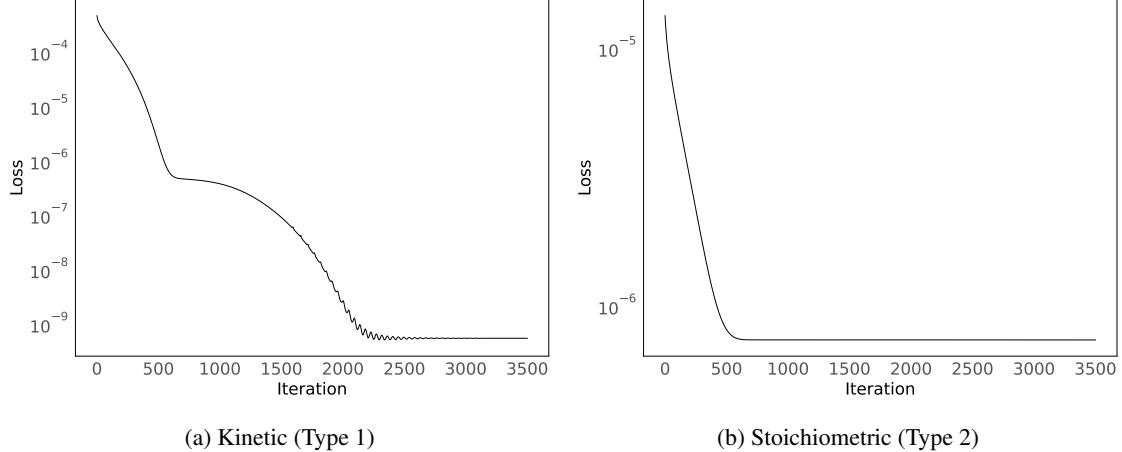


Figure 6: Parameter comparison for white-box two-parameter.



(a) **Kinetic (Type 1):** data is from ground-truth kinetic simulation; WB ansatz is also kinetic simulation.
(b) **Stoichiometric (Type 2):** data is from ground-truth stoichiometric simulation; WB ansatz is also stoichiometric simulation.

Figure 7: **Gradient landscape for the white-box, two-parameter case.** Kinetic (7a) vs stoichiometric (7b) implementation from section 2.1. The stiff gradient vectorfield leads to some degree of parameter nonidentifiability.



(a) Kinetic (Type 1)

(b) Stoichiometric (Type 2)

Figure 8: **Convergence of the training to the final parameter estimates for the white-box, two-parameter case.** Kinetic (8a) vs stoichiometric (8b) implementation from section 2.1. See also fig. 18.

3.3.2 Known Model, Five Unknown Parameters

Next, we repeated the parameter estimation experiment of the previous section but now with five, rather than two, unknown α values. We find that the numerical values are again recovered reasonably well (fig. 9), and with loss dynamics (fig. 10) similar to the two-parameter case.

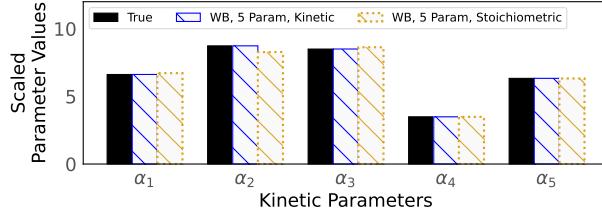


Figure 9: Parameter comparison for the white-box, five-parameter case.

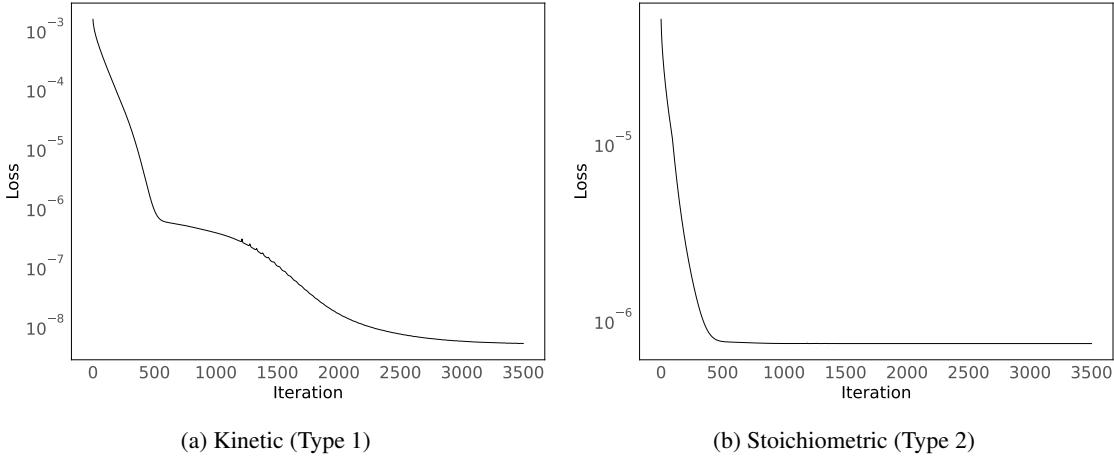


Figure 10: **Convergence of the training to the final parameter estimates for the white-box, five-parameter case.** Kinetic (10a) vs stoichiometric (10b) implementation from section 2.1. See also fig. 19.

3.4 Partially Known Model: Gray-box Identification

For the work in this section, we assumed the expression of $\hat{v}_{I,2}$ in f_{kin} was not known: instead, we only knew that it is a function of GLC and LAC, and we replaced it by a 2-8-8-1 multi-layer perceptron (MLP) with trainable weights and biases. We embedded this MLP into our gray-box computation graph visualized in fig. 1 to make predictions for loss evaluation.

For all four of these experiments, the dataset consisted of 800 single-Euler-step “trajectories” each 0.05 hours in length. The network ansatz was also Euler with a step size of 0.05. Training was 500 epochs of RMSProp with 20 batches per epoch.

3.4.1 Partially Known Model, All Parameters Known

For our first gray-box experiments we further assumed that the values of all kinetic parameters used in expressions beyond that for $\hat{v}_{I,2}$ are correct and do not need to be calibrated. We performed the experiment twice: once with kinetic-based data (data from type-1 simulations) and with the white portion of our gray-box also based on the kinetic formulation; and once with stoichiometric data/version respectively.

For each experiment, we find (fig. 11a and fig. 12a, resp.) that the learned flux functions are largely reproduced correctly, but there are discrepancies (relatively flat network predictions) over some parts of their domain. The greatest percent discrepancy (given in fig. 11b and fig. 12b, scaled by the spread of true values across the training data) arises, as one might expect, at locations where the flux is approximately zero. The most obvious explanation for this error is that this region (small GLC) is not frequently visited by the ground-truth dynamics used for training data, especially in the stoichiometric case. Finally, errors in small fluxes lead have less egregious consequences in what we actually minimize in the network: the prediction error for the concentration evolution.

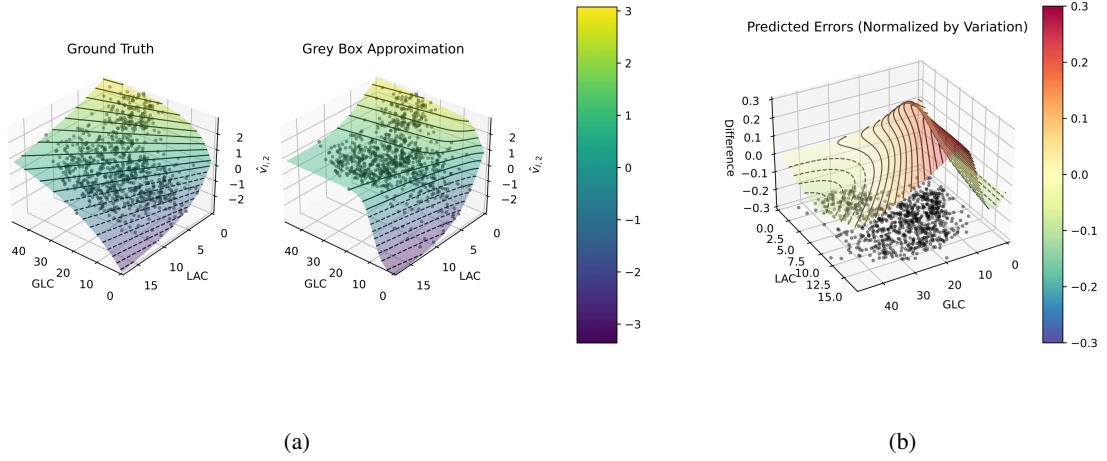


Figure 11: Comparison of fluxes (ground-truth vs kinetic gray-box model). Note that the (GLC, LAC) points visited in training are scattered on the surfaces (11a) or on the base plane (11b). **11a:** ground-truth and neural net approximations of the fluxes given the inputs of the neural net (GLC and LAC). **11b:** normalized prediction errors (fraction of max-min of the true function, across the data). Note the relative rotation (for visual clarity) between (11a) and (11b).

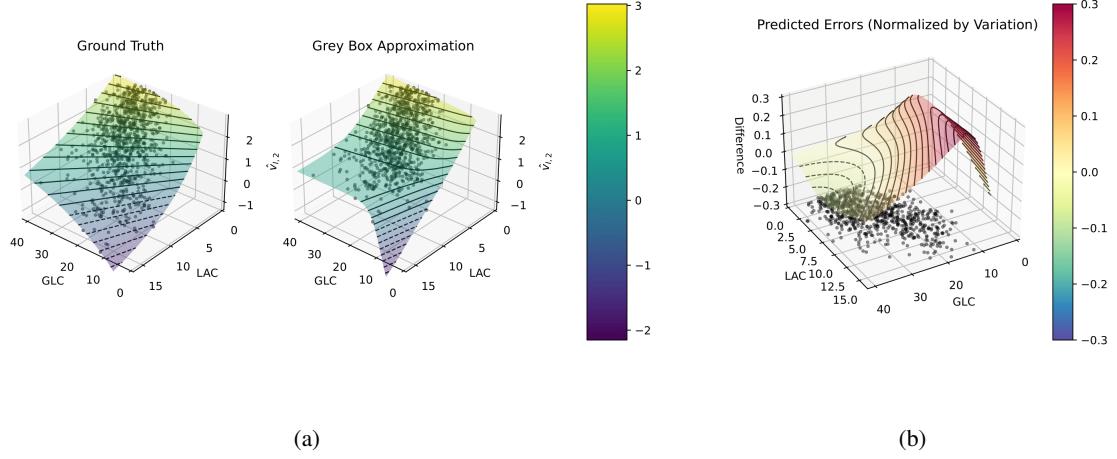


Figure 12: Comparison of fluxes (ground-truth vs stoichiometric gray-box model). **12a:** ground-truth and neural net approximations of the fluxes given the neural net inputs, GLC and LAC. **12b:** normalized prediction errors (fraction of max-min of the true function, across the data.) Note that the (GLC, LAC) points visited in training are scattered on the surfaces (12a) or on the base plane (12b). Note again the relative rotation (for visual clarity) between (12a) and (12b).

3.4.2 Partially Known Model, Partially Known Parameters

Finally, we combine the physical-parameter (white-box) fitting of section 3.3 with the gray-box fitting of section 3.4.1 to produce a model in which we train both neural and physical components jointly. For this experiment, we still assume that the expression for $\hat{v}_{I,2}$ in f_{kin} is unknown; but we also additionally assume the value of one kinetic parameter, α_1 , also needs to be calibrated. As before, we also studied kinetic and stoichiometric versions of the experiment.

As we can see in fig. 13, fig. 14 and table 4 the recovery of the shape of the flux function has similar characteristics to section 3.4.1; yet we are also able to rediscover the parameter value accurately, demonstrating the method’s potential in joint learning with such mixed physical prior information.

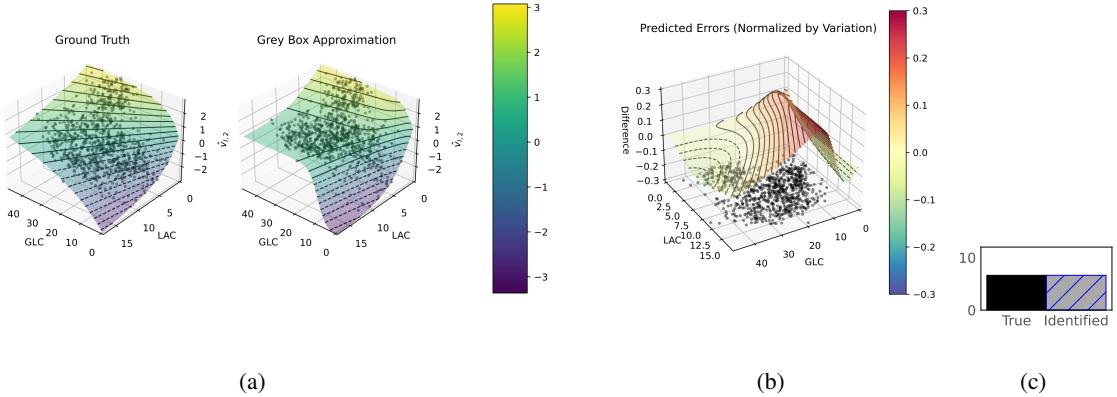


Figure 13: Comparison of fluxes (kinetic ground-truth vs gray-box model). **13a:** ground-truth and neural net approximations of the fluxes as functions of the learned kinetic expression inputs, GLC and LAC. **13b:** normalized (as in fig. 11b and 12b) predicted errors. Black data points plotted as in fig. 11 and 12. **13c:** Ground-truth value and the recovered value of the kinetic parameter α_1 .

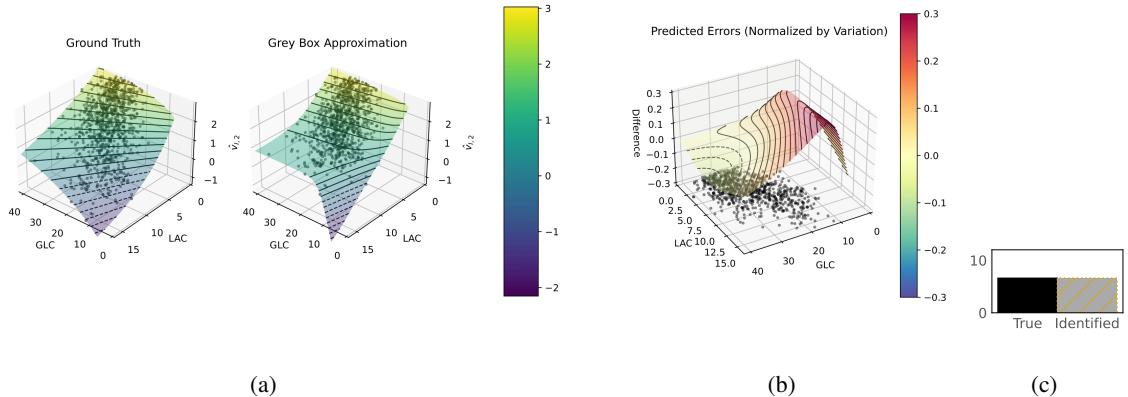


Figure 14: Comparison of fluxes (stoichiometric ground-truth vs gray-box model). **14a:** ground-truth and neural net approximations of the fluxes as functions of the learned kinetic expression inputs GLC and LAC. **14b:** normalized (as in fig. 11b and 12b) predicted errors. Black data points plotted as in fig. 11 and 12. **14c:** Ground-truth value and the recovered value of the kinetic parameter α_1 .

4 Conclusions and Future Directions

In this paper, we revisited a mechanistic model of the biochemical reactions arising in Chinese Hamster Ovary (CHO) cell cultures. When simulating the dynamics of the model, evaluation of the temporal derivative of this system of equations practically necessitates the solution of a constrained convex problem at each time step. This “inner optimization” can lead to: (a) discontinuities in the second time derivatives of the evolving concentrations (that is, the solution itself is C1 as shown in fig. 3); and (b) difficulties in computing the system Jacobian, or sensitivity gradients of evolving states with respect to system parameters.

We then demonstrated how to incorporate such mechanistic physical knowledge of the model along with data-driven approaches, so as to identify or calibrate this type of systems. Our hybrid model can be black-, gray-, or white-box, depending on the portion of physical laws one is confident about *a priori*. Importantly, we implemented a modification of traditional neural network/ODE-net architecture in our white- and gray-box models based on [31]: this approach can encode the differentiable convex optimization layer within a numerical integrator (which can be considered as an unrolled recurrent neural network) so as to overcome the obstacles of computing model gradients. The potential of

this type of data-driven models to identify metabolic network dynamics from data, and perform regression tasks, was illustrated.

The approaches and model architectures that we designed and implemented in this paper should be of broad applicability in fields of engineering where the right-hand-side of the evolution equations intrinsically involves an optimization problem; robotics control, or differentiable Model Predictive Control [44] come to mind. In the metabolic engineering domain, such algorithms can be usefully combined with downstream optimization problems, for the design of experiments, the optimization of feed media composition, or the design of optimal feeding/harvesting policies in bioreactor operation.

Acknowledgements: This work was partially supported by AMBIC. The work of T.B., T.C. and I.G.K. was also partially supported by an AFOSR MURI. The work of C. M. was partially enabled by the DOE Center for Advanced Bioenergy and Bioproducts Science (U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research under Award Number DE-SC0018420) and by the DOE Office of Science, Office of Biological and Environmental Research under Award Number DE-SC0018260).

References

- [1] Michael Butler. Animal cell cultures: recent achievements and perspectives in the production of biopharmaceuticals. *Applied microbiology and biotechnology*, 68:283–291, 2005.
- [2] Ningning Ma, JoAnn Ellet, Centy Okediadi, Paul Hermes, Ellen McCormick, and Susan Casnocha. A single nutrient feed supports both chemically defined ns0 and cho fed-batch processes: Improved productivity and lactate metabolism. *Biotechnology progress*, 25(5):1353–1363, 2009.
- [3] Costas D Maranas and Ali R Zomorrodi. *Optimization methods in metabolic networks*. John Wiley & Sons, 2016.
- [4] Gregory N Stephanopoulos, Aristos A Aristidou, and Jens Nielsen. *Metabolic Engineering: Principles and Methodologies*. Academic Press, October 1998.
- [5] Brett A Boghigian, Gargi Seth, Robert Kiss, and Blaine A Pfeifer. Metabolic flux analysis and pharmaceutical production. *Metabolic engineering*, 12(2):81–95, 2010.
- [6] Chetan Goudar, Richard Biener, C Boisart, Rüdiger Heidemann, James Piret, Albert de Graaf, and Konstantin Konstantinov. Metabolic flux analysis of cho cells in perfusion culture by metabolite balancing and 2d [¹³C, ¹H] cosy nmr spectroscopy. *Metabolic engineering*, 12(2):138–149, 2010.
- [7] Lake-Ee Quek, Stefanie Dietmair, Jens O Krömer, and Lars K Nielsen. Metabolic flux analysis in mammalian cell culture. *Metabolic engineering*, 12(2):161–171, 2010.
- [8] Christophe Chassagnole, Naruemol Noisommit-Rizzi, Joachim W Schmid, Klaus Mauch, and Matthias Reuss. Dynamic modeling of the central carbon metabolism of escherichia coli. *Biotechnology and bioengineering*, 79(1):53–73, 2002.
- [9] Radhakrishnan Mahadevan, Jeremy S Edwards, and Francis J Doyle. Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical journal*, 83(3):1331–1340, 2002.
- [10] Ryan P Nolan and Kyongbum Lee. Dynamic model of CHO cell metabolism. *Metabolic Engineering*, 13(1):108–124, 2011.
- [11] Patrick F Suthers, Charles J Foster, Debolina Sarkar, Lin Wang, and Costas D Maranas. Recent advances in constraint and machine learning-based metabolic modeling by leveraging stoichiometric balances, thermodynamic feasibility and kinetic law formalisms. *Metabolic engineering*, 63:13–33, 2021.
- [12] RA Adomaitis, IG Kevrekidis, RM Farber, AS Lapedes, JL Hudson, and M Kube. Application of neural nets to system identification and bifurcation analysis of real world experimental data. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1990.
- [13] JL Hudson, M Kube, RA Adomaitis, IG Kevrekidis, AS Lapedes, and RM Farber. Nonlinear signal processing and system identification: applications to time series from electrochemical reactions. *Chemical Engineering Science*, 45(8):2075–2081, 1990.
- [14] R Rico-Martines, IG Kevrekidis, MC Kube, and JL Hudson. Discrete-vs. continuous-time nonlinear signal processing: Attractors, transitions and parallel implementation issues. In *1993 American Control Conference*, pages 1475–1479. IEEE, 1993.
- [15] J Nathan Kutz. *Data-Driven Modeling and Scientific Computation: Methods for Complex Systems and Big Data*. Oxford University Press, Inc., USA, 2013.

- [16] Steven L Brunton and J Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2 edition, 2022.
- [17] K Krischer, IG Kevrekidis, MC Kube, and JL Hudson. Discrete-vs. continuous-time nonlinear signal processing of cu electrodissolution data. *Chemical Engineering Communications*, 118(1):25–48, 1992.
- [18] R Rico-Martinez, J S Anderson, and I G Kevrekidis. Continuous-time nonlinear signal processing: a neural network based approach for gray box identification. In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pages 596–605, 1994.
- [19] Cristina P Martin-Linares, Yorgos M Psarellis, Georgios Karapetsas, Eleni D Koronaki, and Ioannis G Kevrekidis. Physics-agnostic and physics-infused machine learning for thin films flows: modeling, and predictions from small data. *arXiv preprint arXiv:2301.12508*, 2023.
- [20] Yorgos M Psarellis, Seungjoon Lee, Tapomoy Bhattacharjee, Sujit S Datta, Juan M Bello-Rivas, and Ioannis G Kevrekidis. Data-driven discovery of chemotactic migration of bacteria via machine learning. *arXiv preprint arXiv:2208.11853*, 2022.
- [21] Felix P Kemeth, Sergio Alonso, Blas Echebarria, Ted Moldenhawer, Carsten Beta, and Ioannis G Kevrekidis. Black and gray box learning of amplitude equations: Application to phase field systems. *arXiv e-prints*, pages arXiv–2207, 2022.
- [22] Felix P Kemeth, Tom Bertalan, Thomas Thiem, Felix Dietrich, Sung Joon Moon, Carlo R Laing, and Ioannis G Kevrekidis. Learning emergent partial differential equations in a learned emergent space. *Nature Communications*, 13(1):3318, 2022.
- [23] Seungjoon Lee, Yorgos M Psarellis, Constantinos I Siettos, and Ioannis G Kevrekidis. Learning black-and gray-box chemotactic pdes/closures from agent based monte carlo simulation data. *arXiv e-prints*, pages arXiv–2205, 2022.
- [24] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [25] Alireza Yazdani, Lu Lu, Maziar Raissi, and George Em Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLOS Computational Biology*, 16(11):1–19, 11 2020.
- [26] Mitchell Daneker, Zhen Zhang, George Em Karniadakis, and Lu Lu. Systems biology: Identifiability analysis and parameter identification via systems-biology informed neural networks. *arXiv preprint arXiv:2202.01723*, 2022.
- [27] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.
- [28] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [29] Paul I Barton and Cha Kun Lee. Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Trans. Model. Comput. Simul.*, 12(4):256–289, oct 2002.
- [30] J A Gomez, K Höffner, and P I Barton. Dfbalab: a fast and reliable matlab code for dynamic flux balance analysis. *BMC Bioinformatics*, 15(409), 2014.
- [31] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, 2017.
- [32] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.
- [33] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.
- [34] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, pages 9558–9570, 2019.
- [35] A Agrawal, S Barratt, S Boyd, E Busseti, and W Moursi. Differentiating through a cone program. *Journal of Applied and Numerical Optimization*, 1(2):107–115, 2019.
- [36] A Agrawal, S Barratt, S Boyd, E Busseti, and W Moursi. diffcp: differentiating through a cone program, version 1.0.19. <https://github.com/cvxgrp/diffcp>, 2019.

- [37] Brandon Amos. *Differentiable Optimization-Based Modeling for Machine Learning*. PhD thesis, Carnegie Mellon University, May 2019.
- [38] Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS’16, page 4608–4616, Red Hook, NY, USA, 2016. Curran Associates Inc.
- [39] E Hairer, S P Nørsett, and G Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, Berlin, Heidelberg, 1993.
- [40] P.I. Barton. Modeling, simulation and sensitivity analysis of hybrid systems. In *CACSD. Conference Proceedings. IEEE International Symposium on Computer-Aided Control System Design (Cat. No.00TH8537)*, pages 117–122, 2000.
- [41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [42] Bryan C Daniels, Yan-Jiun Chen, James P Sethna, Ryan N Gutenkunst, and Christopher R Myers. Sloppiness, robustness, and evolvability in systems biology. *Current Opinion in Biotechnology*, 19(4):389–395, 2008. Protein technologies / Systems biology.
- [43] Alexander Holiday, Mahdi Kooshkbaghi, Juan M. Bello-Rivas, C. William Gear, Antonios Zagaris, and Ioannis G. Kevrekidis. Manifold learning for parameter reduction. *Journal of Computational Physics*, 392:419–431, 2019.
- [44] Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J. Zico Kolter. Differentiable mpc for end-to-end planning and control. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

A Physical Variables and Kinetic Parameters

Variables	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Abbreviations	BIOM	ANTI	GLC	LAC	ALA	ASN	ASP
Names	Biomass	Antibody	Glucose	Lactate	Alanine	Asparagine	Aspartate
Units	mmol/L	mmol/L	mmol/L	mmol/L	mmol/L	mmol/L	mmol/L
Sample Values	1.18	0.0801	30.9	1.36	0.354	5.23	1.21
Kinetic dC_i/dt	2.02	0.0361	-1.76	1.23	0.179	-0.152	-0.097
Stoichiometric dC_i/dt	2.04	0.0668	-1.83	1.2	0.142	-0.153	-0.151
Kinetic - Stoichiometric	-0.0116	-0.0307	0.07	0.0336	0.0372	0.00153	0.0536
Sample Values	0.332	4.34	1.14	7.74	3.17	2.68	0.483
Kinetic dC_i/dt	-0.00657	-0.997	1.3	-0.904	0.181	0.482	0.875
Stoichiometric dC_i/dt	-0.0274	-0.999	1.05	-0.764	0.163	0.679	0.88
Kinetic - Stoichiometric	0.0208	0.00239	0.242	-0.14	0.0179	-0.197	-0.00501

Table 3: Names and units for all variables $\mathbf{C} = (C_1, C_2, \dots, C_{14})^T$.

Numbered Symbol	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9
Physical Symbol	v_{max1}	K_{i1}	K_{m1}	v_{max2f}	K_{m2glc}	v_{max3f}	v_{max3r}	K_{m3ala}	K_{m3glc}
Values (True)	6617.8	87.349	84.982	3490.4	6.3331	950.80	949.28	0.2165	2.0026
§3.3.1 Values (WB 2P Kin)	6617.2	87.354							
Values (WB 2P Sto)	6643.8	81.666							
§3.3.2 Values (WB 5P Kin)	6621	87.371	85.025	3489.9	6.3337				
Values (WB 5P Sto)	6720.2	82.774	86.327	3487	6.3236				
§3.4.2 Values (GB Kin)	6617.9								
Values (GB Sto)	6594.1								
Units	mmol/d	mmol/L	mmol/L	mmol/d	mmol/L	mmol/d	mmol/d	mmol/L	mmol/L
Numbered Symbol	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
Physical Symbol	v_{max8f}	v_{max8r}	K_{m8gln}	K_{m8glu}	K_{m8nh3}	v_{max9f}	v_{max9r}	K_{m9nh3}	v_{max10f}
Values (True)	9568.5	415.83	5.9198	2.0582	7.5053	3.3291	7.8132	0.6866	143.32
Units	mmol/d	mmol/d	mmol/L	mmol/L	mmol/L	mmol/d	mmol/d	mmol/L	mmol/d
Numbered Symbol	α_{19}	α_{20}	α_{21}	α_{22}	α_{23}	α_{24}	α_{25}	α_{26}	α_{27}
Physical Symbol	v_{max10r}	K_{m10asn}	K_{m10asp}	K_{m10nh3}	v_{max11}	v_{max12f}	v_{max12r}	K_{m12ser}	K_{m12gly}
Values (True)	95.194	0.0157	3.5060	0.6301	0.5465	0.6330	92.978	3.0862	0.2020
Units	mmol/d	mmol/L	mmol/d	mmol/L	mmol/d	1	mmol/d	mmol/L	mmol/L
Numbered Symbol	α_{28}	α_{29}	α_{30}	α_{31}	α_{32}	α_{33}	α_{34}	α_{35}	α_{36}
Physical Symbol	v_{max13}	K_{m13}	v_{max16}	K_{m16a}	K_{m16b}	v_{max17}	K_{i17}	v_{max2r}	K_{m2lac}
Values (True)	72.593	1.4396	6316.7	0.9967	0.9901	76.88	46.045	3996.2	4.3040
Units	mmol/d	mmol/L	mmol/d	mmol/L	mmol/L	mmol/d	mmol/L	mmol/d	mmol/L
Numbered Symbol	α_{37}	α_{38}	α_{39}	α_{40}	α_{41}	α_{42}	α_{43}	α_{44}	α_{45}
Physical Symbol	K_{m9glu}	K_{m9gln}	K_{m11asp}	K_{m11asn}	K_{m12nh3}	K_{m16c}	v_{max35}	K_{m35a}	K_{m35b}
Values (True)	0.7519	4.1330	6.4439	6.5433	0.2607	0.0108	2.371	6.8779	9.1087
Units	mmol/L	mmol/L	mmol/L	mmol/L	mmol/L	mmol/L	mmol/d	mmol/L	mmol/L

Table 4: Names and ground-truth values for all parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_{45})^T$, as well as the discovered values of several parameters by white-box and gray-box models. Here, meanings of "2PK" and others can be checked in table 2.

B The ODE Expressions $\frac{dC}{dt} = f_{\text{eqn:ode}}(C; v(C; \alpha))$

$$\begin{aligned}
 \frac{dC_1}{dt} &= v_{18} \frac{C_{14}}{1000} \\
 \frac{dC_2}{dt} &= v_{19} \frac{C_{14}}{1000} \\
 \frac{dC_3}{dt} &= -v_{20} \frac{C_{14}}{1000} \\
 \frac{dC_4}{dt} &= v_{21} \frac{C_{14}}{1000} \\
 \frac{dC_5}{dt} &= v_{22} \frac{C_{14}}{1000} \\
 \frac{dC_6}{dt} &= -v_{23} \frac{C_{14}}{1000} \\
 \frac{dC_7}{dt} &= v_{24} \frac{C_{14}}{1000} \\
 \frac{dC_8}{dt} &= -v_{25} \frac{C_{14}}{1000} \\
 \frac{dC_9}{dt} &= -v_{26} \frac{C_{14}}{1000} \\
 \frac{dC_{10}}{dt} &= v_{27} \frac{C_{14}}{1000} \\
 \frac{dC_{11}}{dt} &= -v_{28} \frac{C_{14}}{1000} \\
 \frac{dC_{12}}{dt} &= v_{29} \frac{C_{14}}{1000} \\
 \frac{dC_{13}}{dt} &= (v_{25} + v_{33}) \frac{C_{14}}{1000} \\
 \frac{dC_{14}}{dt} &= \frac{\frac{dC_1}{dt}}{2.31} - 0.003C_{14}
 \end{aligned}$$

C List of all Reactions

ID	Kin.	Stoi.	Kin. - Stoi.	Reaction
1	1.18	2.22	-1.04	$G6P \rightarrow 2PYR + 3ATP + 2NADH(\text{Cytosolic})$
2	0.0801	-0.901	0.981	$PYR + NADH(\text{Cytosolic}) \leftrightarrow LAC$
3	30.9	28.6	2.32	$PYR + GLU \leftrightarrow ALA + AKG$
4	1.024 $\times 10^{-12}$	1.153 $\times 10^{-15}$	1.023 $\times 10^{-12}$	$PYR + OXA \rightarrow AKG + 2CO_2 + 2NADH(\text{Mitochondrial})$
5	22.5	20.4	2.06	$AKG \rightarrow MAL + CO_2 + NADH(\text{Mitochondrial}) + FADH_2 + ATP$
6	1.151 $\times 10^{-13}$	7.718 $\times 10^{-16}$	1.144 $\times 10^{-13}$	$MAL \rightarrow OXA + NADH(\text{Mitochondrial})$
7	25.7	22.3	3.35	$MAL \rightarrow PYR + CO_2$
8	1.36	1.36	-2.852 $\times 10^{-5}$	$GLN \leftrightarrow GLU + NH_3$
9	0.354	3.78	-3.42	$AKG + NH_3 + NADH(\text{Mitochondrial}) \leftrightarrow GLU$
10	5.23	5.23	-3.530 $\times 10^{-5}$	$ASN \leftrightarrow ASP + NH_3$
11	1.21	1.82	-0.61	$ASP + AKG \leftrightarrow OXA + GLU$
12	0.332	-0.597	0.929	$SER + CO_2 + NH_3 + NADH(\text{Cytosolic}) \leftrightarrow 2GLY$
13	4.34	3.41	0.929	$Cystine + NADH(\text{Cytosolic}) \rightarrow 2Cysteine$
14	14.8	19.4	-4.55	$NADH(\text{Mitochondrial}) + 0.5O_2 \rightarrow 2.5ATP$
15	16.6	20.5	-3.89	$FADH_2 + 0.5O_2 \rightarrow 1.5ATP$
16	1.14	4.27	-3.13	$0.0838ALA + 0.041ASN + 0.0804ASP + 8.6825ATP + 0.0261Cysteine + 0.452G6P + 0.0873GLN + 0.056GLY + 0.427OXA + 0.096SER \rightarrow BIOM + 0.004FADH_2 + 0.0082GLU + 0.4445MAL + 0.6391NADH(\text{Mitochondrial}) + 0.2085PYR$
17	7.74	7.34	0.399	$0.0614ALA + 0.0344ASN + 0.0389ASP + 9.2ATP + 0.024Cysteine + 0.0479GLU + 0.0449GLN + 0.0719GLY + 0.1SER \rightarrow ANTI$
18	3.17	4.27	-1.1	$BIOM \rightarrow BIOM(\text{Extracellular})$
19	2.68	7.34	-4.66	$ANTI \rightarrow ANTI(\text{Extracellular})$
20	4.28	2.89	1.39	$GLC(\text{Extracellular}) + ATP \rightarrow G6P$
21	0.0801	-0.901	0.981	$LAC \leftrightarrow LAC(\text{Extracellular})$
22	30.3	27.8	2.56	$ALA \leftrightarrow ALA(\text{Extracellular})$
23	5.54	5.65	-0.115	$ASN(\text{Extracellular}) \rightarrow ASN$
24	3.62	2.78	0.846	$ASP \leftrightarrow ASP(\text{Extracellular})$
25	5.756 $\times 10^{-15}$	3.41	-3.41	$Cystine(\text{Extracellular}) + GLU \rightarrow Cystine + GLU(\text{Extracellular})$
26	1.8	2.06	-0.255	$GLN(\text{Extracellular}) \leftrightarrow GLN$
27	0.0431	-1.96	2.01	$GLY \leftrightarrow GLY(\text{Extracellular})$
28	0.249	3.07	-2.82	$SER(\text{Extracellular}) \rightarrow SER$
29	5.9	3.4	2.5	$NH_3 \leftrightarrow NH_3(\text{Extracellular})$
30	19.9	23.2	-3.26	$O_2(\text{Extracellular}) \leftrightarrow O_2$
31	47.8	43.4	4.47	$CO_2 \rightarrow CO_2(\text{Extracellular})$
32	4.23	3.27	0.965	$2Cysteine + O_2 \rightarrow 2Cystine$
33	8.539 $\times 10^{-15}$	-22.8	22.8	$GLU \rightarrow GLU(\text{Extracellular})$
34	-5.6 $\times 10^{-15}$	4.368	-5.6	$NADH(\text{Cytosolic}) \rightarrow 0.5NADH(\text{Mitochondrial}) + 0.5FADH_2$
35	0.483	-1.26	1.75	$G6P + ATP + 2GLU \rightarrow 2NADH(\text{Cytosolic}) + 2SER + 2AKG$

Table 5: Expressions of all reactions in the metabolic network. Note that we distinguish between cytosolic and mitochondrial NADH, while all metabolites are intracellular unless indicated otherwise. Reactions with “ \leftrightarrow ” are reversible, while “ \rightarrow ” is used if and only if the reaction is irreversible. ID numbers are colored as described in appendix F. Fluxes (numeric columns) are evaluated at the same state as used in table 3.

D The Kinetic Expressions $\hat{v}_I = f_{\text{kin}}(C; \alpha)$

$$\begin{aligned}
 \hat{v}_{I,1} &= \frac{v_{\max 1} \frac{\text{GLC}}{K_{m1}}}{1 + \frac{\text{LAC}}{K_{i1}} + \frac{\text{GLC}}{K_{m1}} + \frac{\text{LAC}}{K_{i1}} \frac{\text{GLC}}{K_{m1}}} \\
 \hat{v}_{I,2} &= \frac{v_{\max 2f} \frac{\text{GLC}}{K_{m2glc}} - v_{\max 2r} \frac{\text{LAC}}{K_{m2lac}}}{1 + \frac{\text{GLC}}{K_{m2glc}} + \frac{\text{LAC}}{K_{m2lac}}} \\
 \hat{v}_{I,3} &= \frac{v_{\max 3f} \frac{\text{GLC}}{K_{m3glc}} - v_{\max 3r} \frac{\text{ALA}}{K_{m3ala}}}{1 + \frac{\text{GLC}}{K_{m3glc}} + \frac{\text{ALA}}{K_{m3ala}}} \\
 \hat{v}_{I,4} &= \frac{v_{\max 8f} \frac{\text{GLN}}{K_{m8gln}} - v_{\max 8r} \frac{\text{GLU}}{K_{m8glu}} \frac{\text{NH3}}{K_{m8nh3}}}{1 + \frac{\text{GLN}}{K_{m8gln}} + \frac{\text{GLU}}{K_{m8glu}} + \frac{\text{NH3}}{K_{m8nh3}} + \frac{\text{GLU}}{K_{m8glu}} \frac{\text{NH3}}{K_{m8nh3}}} \\
 \hat{v}_{I,5} &= \frac{v_{\max 9f} \frac{\text{NH3}}{K_{m9nh3}} - v_{\max 9r} \left(\frac{\text{GLU}}{K_{m9glu}} + \frac{\text{GLN}}{K_{m9gln}} \right)}{1 + \frac{\text{NH3}}{K_{m9nh3}} + \frac{\text{GLU}}{K_{m9glu}} + \frac{\text{GLN}}{K_{m9gln}} + \frac{\text{GLU}}{K_{m9glu}} \frac{\text{GLN}}{K_{m9gln}}} \\
 \hat{v}_{I,6} &= \frac{v_{\max 10f} \frac{\text{ASN}}{K_{m10asn}} - v_{\max 10r} \frac{\text{ASP}}{K_{m10asp}} \frac{\text{NH3}}{K_{m10nh3}}}{1 + \frac{\text{ASN}}{K_{m10asn}} + \frac{\text{ASP}}{K_{m10asp}} + \frac{\text{NH3}}{K_{m10nh3}} + \frac{\text{ASP}}{K_{m10asp}} \frac{\text{NH3}}{K_{m10nh3}}} \\
 \hat{v}_{I,7} &= \frac{v_{\max 11} \left(\frac{\text{ASP}}{K_{m11asp}} + \frac{\text{ASN}}{K_{m11asn}} \right)}{1 + \frac{\text{ASN}}{K_{m11asn}} + \frac{\text{ASP}}{K_{m11asp}} + \frac{\text{ASP}}{K_{m11asp}} \frac{\text{ASN}}{K_{m11asn}}} \\
 \hat{v}_{I,8} &= \frac{v_{\max 12f} \cdot \hat{v}_{I,10} \frac{\text{SER}}{K_{m12ser}} \frac{\text{NH3}}{K_{m12nh3}} - v_{\max 12r} \left(\frac{\text{GLY}}{K_{m12gly}} \right)^2}{1 + \frac{\text{SER}}{K_{m12ser}} + \frac{\text{GLY}}{K_{m12gly}} + \frac{\text{NH3}}{K_{m12nh3}} + \left(\frac{\text{GLY}}{K_{m12gly}} \right)^2} \\
 \hat{v}_{I,9} &= \frac{v_{\max 13} \frac{\text{CYS}}{K_{m13}}}{1 + \frac{\text{CYS}}{K_{m13}}} \\
 \hat{v}_{I,10} &= \frac{v_{\max 16} \frac{\text{GLN}}{K_{m16a}} \frac{\text{ASN}}{K_{m16b}} \frac{\text{ALA}}{K_{m16c}}}{1 + \frac{\text{GLN}}{K_{m16a}} + \frac{\text{ASN}}{K_{m16b}} + \frac{\text{ALA}}{K_{m16c}} + \frac{\text{GLN}}{K_{m16a}} \frac{\text{ASN}}{K_{m16b}} + \frac{\text{GLN}}{K_{m16a}} \frac{\text{ALA}}{K_{m16c}} + \frac{\text{ASN}}{K_{m16b}} \frac{\text{ALA}}{K_{m16c}} + \frac{\text{GLN}}{K_{m16a}} \frac{\text{ASN}}{K_{m16b}} \frac{\text{ALA}}{K_{m16c}}} \\
 \hat{v}_{I,11} &= \frac{v_{\max 17}}{1 + \frac{\text{LAC}}{K_{i17}}} \\
 \hat{v}_{I,12} &= \hat{v}_{I,10} \\
 \hat{v}_{I,13} &= \hat{v}_{I,11} \\
 \hat{v}_{I,14} &= \frac{v_{\max 35} \frac{\text{GLC}}{K_{m35a}} \left(\frac{\text{GLU}}{K_{m35b}} \right)^2}{1 + \frac{\text{GLC}}{K_{m35a}} + \frac{\text{GLU}}{K_{m35b}} + \left(\frac{\text{GLU}}{K_{m35b}} \right)^2}
 \end{aligned}$$

E Deviations in Kinetic Expressions from [10]

All kinetically defined intracellular reactions have been modeled using the Michaelis Menten format, based on kinetic expressions from Nolan and Lee [10]. Some of the modifications are summarized below and will be described in detail in a separate publication (Khare et al., to be published).

The temperature dependent constants TC which were used to scale the maximum forward reaction rate - v_{max} , have been removed from the kinetic expressions as have the inhibition exponential constants exp_i . Secondly, the redox variable R which was used to account for the redox state of the cell in the reaction kinetics was removed. Instead, reactions were modeled solely based on the extracellular metabolites with consumption and production rates being dependent on the concentrations of the associated extracellular metabolites. Reaction stoichiometries were revised and additional reactions were included based on literature and ^{13}C -labeled (intracellular and extracellular) metabolite tracking data. Reasons for the modifications are listed below:

- Reaction 1 (corresponding to $\hat{v}_{I,1}$): $\text{G6P} \rightarrow 2\text{PYR} + 3\text{ATP} + 2\text{NADH}$
Temperature dependent constant TC_1 and inhibition exponential constant exp_1 (which is also temperature dependent) were removed from the kinetic expression.
- Reaction 2 (corresponding to $\hat{v}_{I,2}$): $\text{PYR} + \text{NADH} \leftrightarrow \text{LAC}$
Redox variable R removed and reaction made continuous.
- Reaction 3 (corresponding to $\hat{v}_{I,3}$): $\text{PYR} + \text{GLU} \leftrightarrow \text{ALA} + \text{AKG}$
Temperature constant TC_{3b} removed.
- Reaction 8 (corresponding to $\hat{v}_{I,4}$): $\text{GLN} \leftrightarrow \text{GLU} + \text{NH}_3$
Temperature constant TC_8 removed.
- Reaction 9 (corresponding to $\hat{v}_{I,5}$): $\text{AKG} + \text{NH}_3 + \text{NADH} \leftrightarrow \text{GLU}$
Dependence on intracellular metabolite AKG (not dynamically tracked) removed, to make reaction expression adhere to Michaelis Menten kinetics.
- Reaction 10 (corresponding to $\hat{v}_{I,6}$): $\text{ASN} \leftrightarrow \text{ASP} + \text{NH}_3$
Temperature dependent constant TC_{10} removed from kinetic expression.
- Reaction 11 (corresponding to $\hat{v}_{I,7}$): $\text{ASP} + \text{AKG} \leftrightarrow \text{OXA} + \text{GLU}$
The reaction expression has been changed to be irreversible.
- Reaction 12 (corresponding to $\hat{v}_{I,8}$): $\text{SER} + \text{CO}_2 + \text{NH}_3 + \text{NADH} \leftrightarrow 2\text{GLY}$
Forward reaction made dependent on ammonia in addition to serine.
- Reaction 13 (corresponding to $\hat{v}_{I,9}$): $\text{Cystine} + \text{NADH} \rightarrow 2\text{Cysteine}$
Variable r , which is temperature dependent, has been removed.
- Reaction 16 (corresponding to $\hat{v}_{I,10}$): Biomass production
Temperature constant TC_{16} and temperature dependency removed.
- Reaction 17 (corresponding to $\hat{v}_{I,11}$): Antibody synthesis
Exponential inhibition constant exp_{17} removed.
- Reaction 33: Removed from our expressions since we have defined expressions for only intracellular reactions as Michaelis-Menten and not transport reactions. The transport reactions are all calculated stoichiometrically.
- Reaction 35 (corresponding to $\hat{v}_{I,14}$): $\text{G6P} + \text{ATP} + 2\text{GLU} \rightarrow 2\text{NADH}(\text{Cytosolic}) + 2\text{SER} + 2\text{AKG}$
This reaction was added based on the presence of the serine synthesis pathway (SSP) originating from glycolysis.

F The Stoichiometric Matrix S

Blue: indices of intracellular fluxes

Olive Overlined: indices of reversible extracellular fluxes

Red Underlined: indices of irreversible extracellular fluxes

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
AKG	0	0	1	-1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ALA	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-0.0838	-0.0614	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0		
ANTI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ASN	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	-0.041	-0.0344	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		
ASP	0	0	0	0	0	0	0	1	-1	0	0	0	0	-0.0804	-0.0389	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0		
ATP	3	0	0	1	0	0	0	0	0	0	2.5	1.5	-8.6825	-9.2	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1		
BIOM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
CO ₂	0	0	0	2	1	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Cysteine	0	0	0	0	0	0	0	0	0	0	2	0	0	-0.0261	-0.024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	0	0		
Cystine	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
FADH ₂	0	0	0	1	0	0	0	0	0	0	0	-1	0.004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	
G6P	-1	0	0	0	0	0	0	0	0	0	0	0	-0.452	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	
GLU	0	0	-1	0	0	0	1	1	0	0	0	0	0.0082	-0.0479	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	-1	0	-2		
GLN	0	0	0	0	0	0	-1	0	0	0	0	0	-0.0873	-0.0449	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0		
GLY	0	0	0	0	0	0	0	0	0	0	2	0	0	-0.056	-0.0719	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0		
LAC	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MAL	0	0	0	1	-1	0	0	0	0	0	0	0	0.4445	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
NADH(Cytosolic)	2	-1	0	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2
NADH(Mitochondrial)	0	0	0	2	1	0	0	-1	0	0	-1	0	0.6391	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	
NH ₃	0	0	0	0	0	0	1	-1	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
O ₂	0	0	0	0	0	0	0	0	0	0	-0.5	-0.5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	
OXA	0	0	0	-1	0	1	0	0	0	1	0	0	-0.427	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
PYR	2	-1	-1	0	1	0	0	0	0	0	0	0	0.2085	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SER	0	0	0	0	0	0	0	0	0	0	-1	0	0	-0.096	-0.1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	

G Closed-form Solution for Stoichiometric-based Approach (w/o Inequality Bounds)

The objective function for Equation eq. (6) could be rewritten as:

$$\|\mathbf{v}_I - \hat{\mathbf{v}}_I\|_2^2 = \mathbf{v}_I^T \cdot \mathbf{v}_I - 2\hat{\mathbf{v}}_I \cdot \mathbf{v}_I + \hat{\mathbf{v}}_I^T \cdot \hat{\mathbf{v}}_I. \quad (19)$$

Notice that the last term of eq. (19) is not the function of \mathbf{v}_I or \mathbf{v}_E , which will be further dropped, and lead to a revised form of the optimization problem:

$$\min_{\mathbf{v}_I, \mathbf{v}_E} \mathbf{v}_I^T \cdot \mathbf{v}_I - 2\hat{\mathbf{v}}_I \cdot \mathbf{v}_I, \text{ s.t. } \mathbf{S}_I \cdot \mathbf{v}_I + \mathbf{S}_E \cdot \mathbf{v}_E = 0. \quad (20)$$

In order to solve this optimization problem, we define the Lagrangian function

$$\mathcal{L} = \mathbf{v}_I^T \cdot \mathbf{v}_I - 2\hat{\mathbf{v}}_I \cdot \mathbf{v}_I + \boldsymbol{\gamma}^T \cdot (\mathbf{S}_I \cdot \mathbf{v}_I + \mathbf{S}_E \cdot \mathbf{v}_E), \quad (21)$$

where $\boldsymbol{\gamma} \in \mathbb{R}^M$ is the vector of Lagrange multiplier. A necessary condition for \mathbf{v}_I to be an extremum is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}_I} = 2\mathbf{v}_I - 2\hat{\mathbf{v}}_I + \mathbf{S}_I^T \boldsymbol{\gamma} = 0. \quad (22)$$

A necessary condition for \mathbf{v}_E to be an extremum is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}_E} = \mathbf{S}_E^T \boldsymbol{\gamma} = 0. \quad (23)$$

A necessary condition for $\boldsymbol{\gamma}$ to be an extremum is just the constraint

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\gamma}} = \mathbf{S}_I \cdot \mathbf{v}_I + \mathbf{S}_E \cdot \mathbf{v}_E = 0. \quad (24)$$

Notice that eq. (22), eq. (23) and eq. (24) are linear equations of \mathbf{v}_E , \mathbf{v}_I and $\boldsymbol{\gamma}$, which leads to a linear system of $E + I + M = 21 + 14 + 24 = 59$ equations and variables:

$$\begin{bmatrix} 2\mathbf{I}_I & 0 & \mathbf{S}_I^T \\ 0 & 0 & \mathbf{S}_E^T \\ \mathbf{S}_I & \mathbf{S}_E & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_I \\ \mathbf{v}_E \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} 2\hat{\mathbf{v}}_I \\ 0 \\ 0 \end{bmatrix}, \quad (25)$$

where $\mathbf{I}_I \in \mathbb{R}^{I \times I}$ is the identity matrix. As we computed that the determinant of

$$\mathbf{A} = \begin{bmatrix} 2\mathbf{I}_I & 0 & \mathbf{S}_I^T \\ 0 & 0 & \mathbf{S}_E^T \\ \mathbf{S}_I & \mathbf{S}_E & 0 \end{bmatrix} \quad (26)$$

is nonzero (aka \mathbf{A} is invertible), we know eq. (25) has a unique solution:

$$\begin{bmatrix} \mathbf{v}_I \\ \mathbf{v}_E \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} 2\mathbf{I}_I & 0 & \mathbf{S}_I^T \\ 0 & 0 & \mathbf{S}_E^T \\ \mathbf{S}_I & \mathbf{S}_E & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\hat{\mathbf{v}}_I \\ 0 \\ 0 \end{bmatrix}. \quad (27)$$

If we denote $\mathbf{I}_E \in \mathbb{R}^{E \times E}$ as an identity matrix, then

$$\begin{bmatrix} \mathbf{v}_I \\ \mathbf{v}_E \end{bmatrix} = \begin{bmatrix} \mathbf{I}_I & 0 & 0 \\ 0 & \mathbf{I}_E & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_I \\ \mathbf{v}_E \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_I & 0 & 0 \\ 0 & \mathbf{I}_E & 0 \end{bmatrix} \begin{bmatrix} 2\mathbf{I}_I & 0 & \mathbf{S}_I^T \\ 0 & 0 & \mathbf{S}_E^T \\ \mathbf{S}_I & \mathbf{S}_E & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\hat{\mathbf{v}}_I \\ 0 \\ 0 \end{bmatrix} \quad (28)$$

is the closed-form expression for stoichiometric-based approach (w/o bounds).

H Additional figures

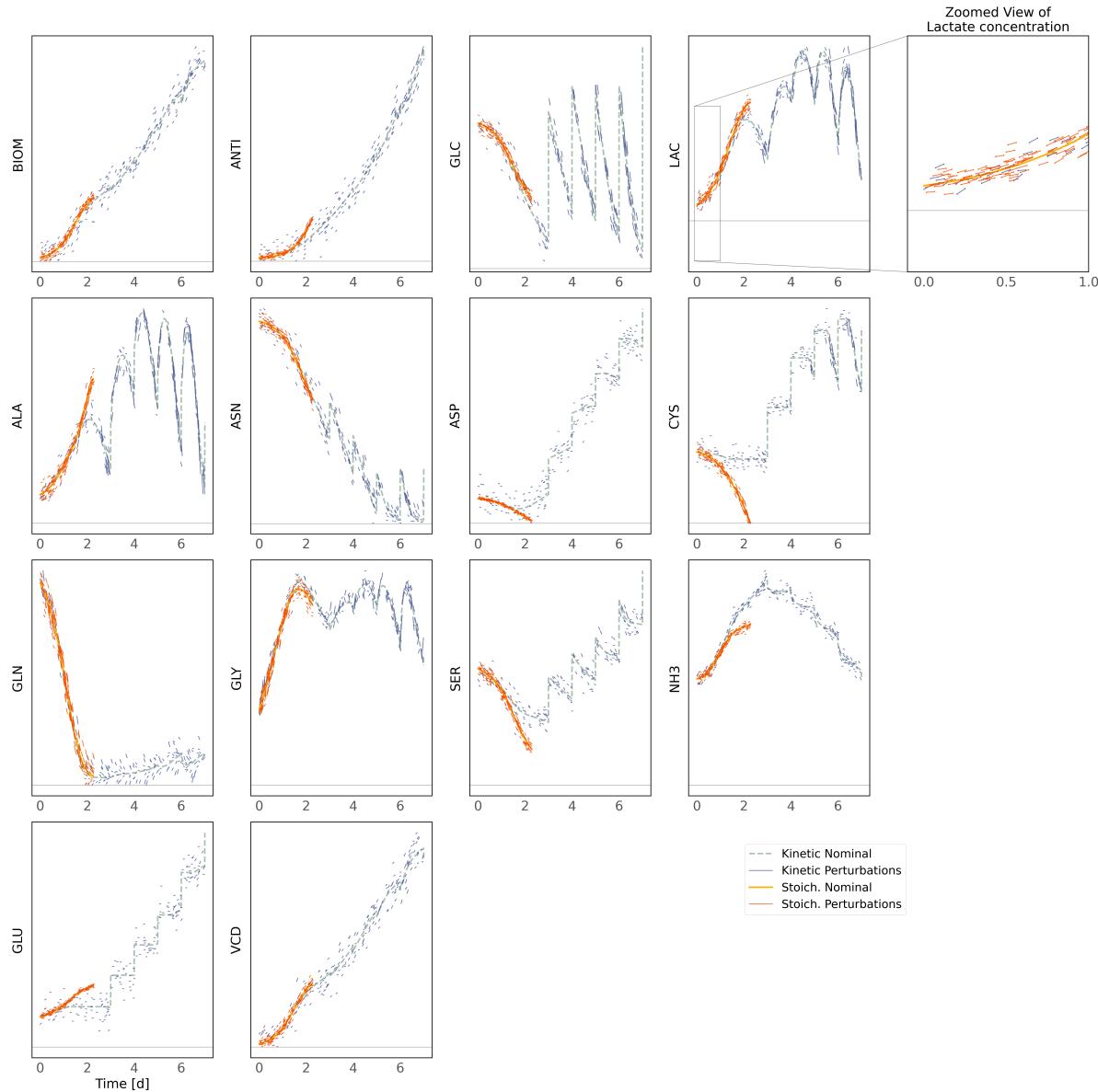


Figure 15: Nominal trajectories overlaid on sampled short-time flows from perturbed initial conditions. Color is used to distinguish between different curves described in the legend.

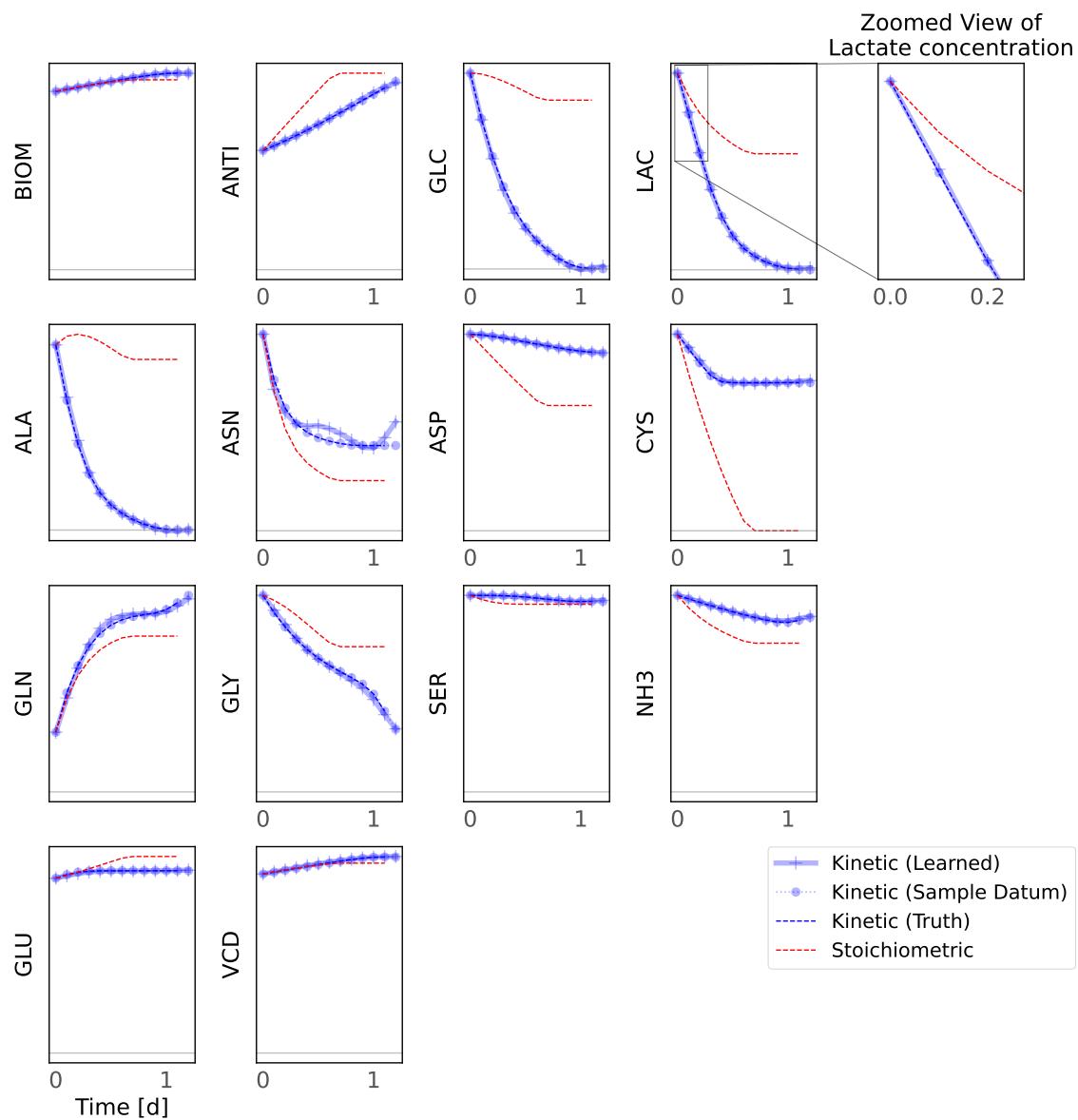


Figure 16: Black-box predictions all variables (Kinetic (Type 1)). See also fig. 4. Color is used to distinguish between different curves described in the legend.

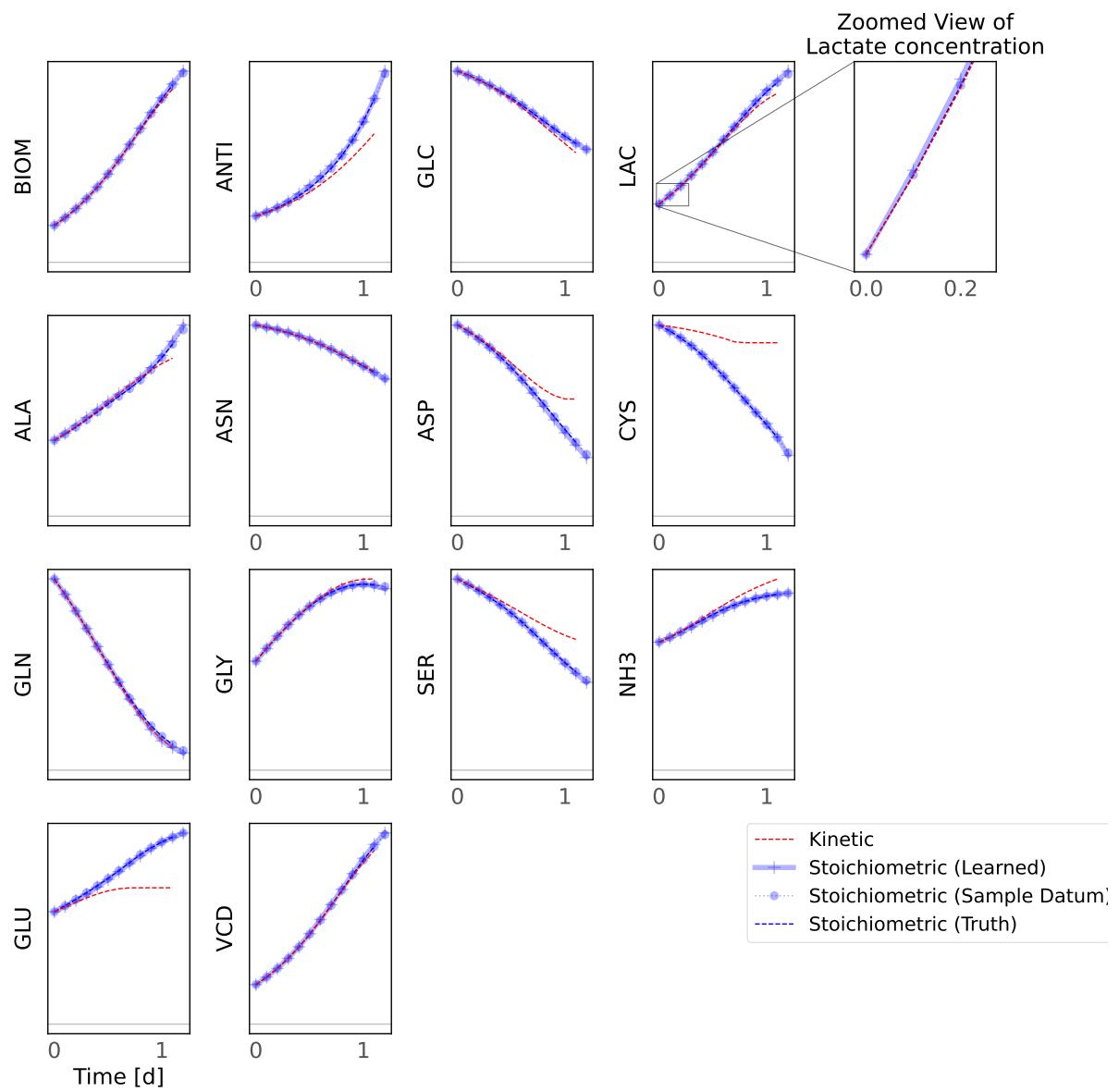


Figure 17: Black-box predictions; all variables (Stoichiometric (Type 2)). Color is used to distinguish between different curves described in the legend.

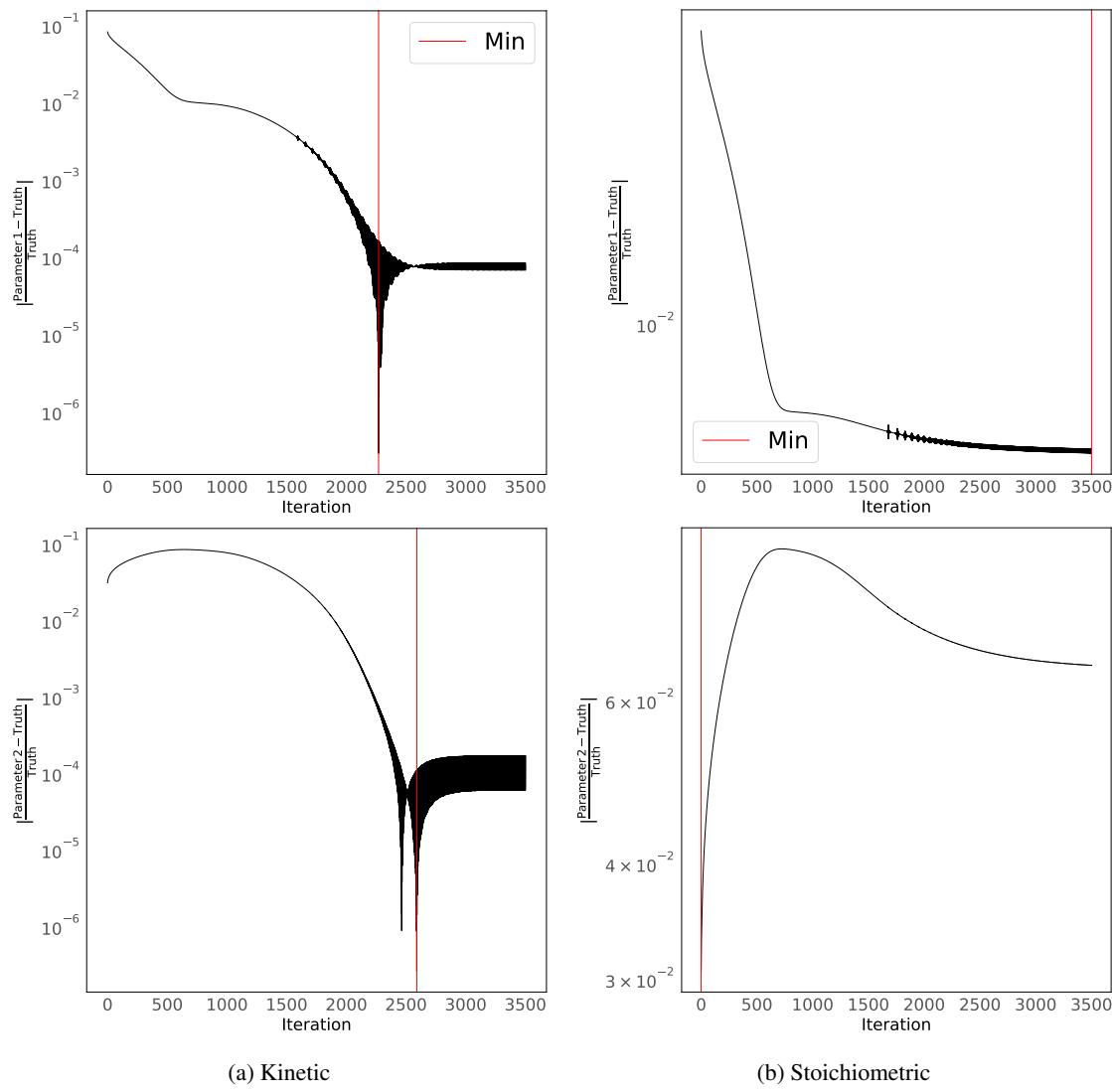


Figure 18: Absolute parameter error history. (Two-parameter white-box case.)

Data-driven and Physics Informed Modelling of Chinese Hamster Ovary Cell Bioreactors

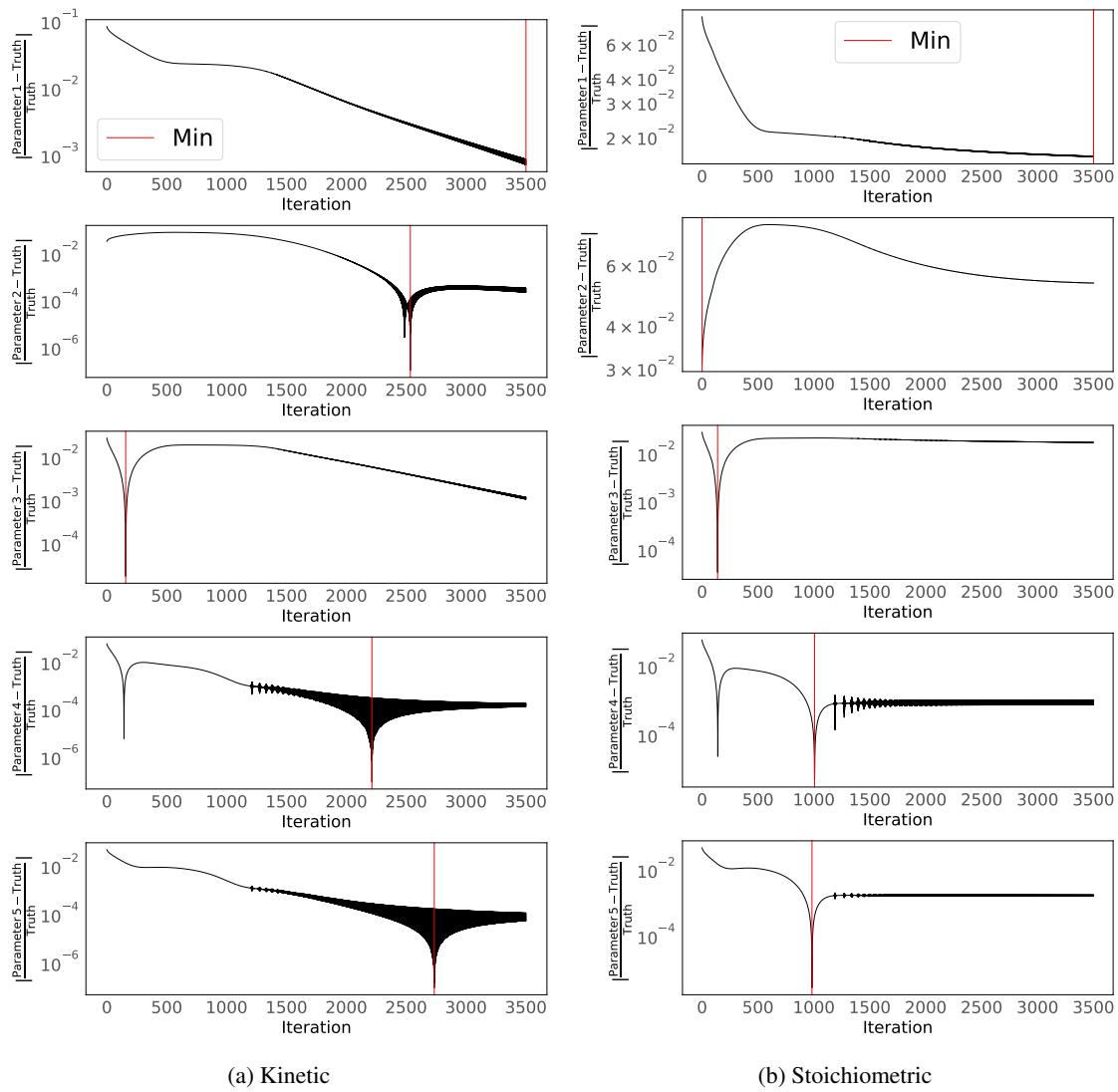


Figure 19: Absolute parameter error history, five-parameter white-box case.