Data-driven and physics informed modeling of Chinese Hamster Ovary cell bioreactors

Tianqi Cui [a,1], Tom Bertalan [a,1], Nelson Ndahiro [a], Pratik Khare [a], Michael Betenbaugh [a], Costas Maranas [b], Ioannis G. Kevrekidis [a,c,*]

[a] *Department of Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, MD 21218, USA*
[b] *Department of Chemical Engineering, The Pennsylvania State University, University Park, PA 16802, USA*
[c] *Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218, USA*

ARTICLE INFO

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 modeling); (b) recover unknown physical parameters ("white-box" parameter fitting) or—importantly—(c) learn partially unknown kinetic expressions (gray-box modeling) 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 (Butler, 2005). 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 (Ma et al., 2009). 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 (Maranas and Zomorrodi, 2016; Stephanopoulos et al., 1998). Metabolic Flux Analysis (MFA) leveraging stable carbon (i.e., 13C) labeled substrates techniques is the only technique that can provide information on internal fluxes (Boghigian et al., 2010;

Goudar et al., 2010; Quek et al., 2010). 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 (Chassagnole et al., 2002; Mahadevan et al., 2002). 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 (Nolan and Lee, 2011). 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

---

* Corresponding author.
  E-mail addresses: tom@tombertalan.com (T. Bertalan), yannisk@jhu.edu (I.G. Kevrekidis).
[1] These authors contributed equally.

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 (Suthers et al., 2021).

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 (Adomaitis et al., 1990; Hudson et al., 1990; Rico-Martines et al., 1993; Kutz, 2013; Brunton and Kutz, 2022) 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 (Hudson et al., 1990; Adomaitis et al., 1990; Krischer et al., 1992; Rico-Martinez et al., 1994; Rico-Martines et al., 1993). They can also be used to directly infer the evolution of the system variables when underlying physics are unclear (Martin-Linares et al., 2023; Psarellis et al., 2022; Kemeth et al., 2022a,b; Lee et al., 2022). Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019), Systems-Biology-Informed Neural Networks (SBINNs) (Yazdani et al., 2020; Daneker et al., 2022), and similar architectures (Lu et al., 2021), can and have been used to solve supervised learning tasks while respecting known laws of physics, system biology, et al. (Karniadakis et al., 2021). 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 Appendices D and E,) to the model presented in Nolan and Lee (2011), which constitutes our starting point) describes a continuous-time dynamical system (the terms are defined in Table 1):

$$\frac{\mathrm{d}C}{\mathrm{d}t} = f_{\mathbf{eqn:ode}}(C; v(C; \alpha));$$ (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 (Barton and Lee, 2002; Gomez et al., 2014; Mahadevan et al., 2002) we will here refer to these as **D**ynamic **F**lux **B**alance **A**nalysis (DFBA) equations.

Here, $C \in \mathbb{R}^K$ ($K = 14$) 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 4); $v \in \mathbb{R}^N$ ($N = E + I = 35$) are all fluxes (reaction rates, see Appendix C for all reaction expressions) including I intracellular fluxes (which can be precomputed from the C) $v_I \in \mathbb{R}^I$ ($I = 14$) and E extracellular fluxes $v_E \in \mathbb{R}^E$ ($E = 21$). Some of extracellular fluxes are assumed to be irreversible ($v_{E,ir} \in \mathbb{R}^{E_{ir}}$ ($E_{ir} = 14$), $v_{E,ir} \geq 0$), while others are assumed reversible ($v_{E,r} \in \mathbb{R}^{E_r}$ ($E_r = 7$)); v is a function of C and α, where $\alpha \in \mathbb{R}^P$ ($P = 45$) are the kinetic parameters.

Given C and α, the evaluation of $f_{\mathbf{eqn:ode}}$ in Eq. (1) is typically done in one of two very different ways. Both involve the following steps,

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

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

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

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

   where $f_{\mathbf{kin}} : \mathbb{R}^{K \times P} \mapsto \mathbb{R}^I$ (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 Nolan and Lee (2011)).

2. The fluxes have to satisfy some constraints:

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

     $$S \cdot v = 0,$$ (3)

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

     $$S_I \cdot v_I + S_E \cdot v_E = 0$$

   - Among all 21 extracellular fluxes v_E, 14 of them are known to be irreversible, which requires

     $$v_{E,ir} = B_{ir} \cdot v_E \geq 0,$$ (4)

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

Notice that the combination of Equations Eqs. (2) and (3) consists of 38 independent linear equations, while the unknown variable 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 v_I and extracellular 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

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

(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 $\boldsymbol{v}_E$ by computing the pseudo inverse of $\mathbf{S}_E$:

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

(b) The stoichiometric-based approach (Nolan and Lee, 2011): we satisfy the stoichiometric equations exactly Eq. (3), and then approximately satisfy the kinetic equations Eq. (2), which leads to

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

(6)

This is a least squares optimization problem with linear constraints. If we ignored the inequality constraints, we would obtain an analytical solution of $(\boldsymbol{v}_I, \boldsymbol{v}_E)$ 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 $\hat{\boldsymbol{v}}_I$ values: divide them by 1000 to adjust their numerical range to $\sim 1-10$ upon entering either optimization problem, and multiply the resulting fluxes $\boldsymbol{v}_I$ and $\boldsymbol{v}_E$ by the same factor before exiting This does not change the solution, but improves the numerical conditioning.

After the optimization step we have

$$\boldsymbol{v} = \mathbf{B}_I \cdot \boldsymbol{v}_I + \mathbf{B}_E \cdot \boldsymbol{v}_E = \boldsymbol{f}_{\text{optim}}^{(\ell)}(\hat{\boldsymbol{v}}_I),$$

(7)

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

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

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

(8)

As the fluxes $\boldsymbol{v}$ 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,

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

(9)

where $\{t_i : i = 0, 1, 2, \ldots\}$ 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* (Mahadevan et al., 2002) and then proceeding to the more recent mammalian biomanufacturing targets (Nolan and Lee, 2011). Beyond the constraints on our inner (optimization) problem that we showed above, additional constraints were imposed in Mahadevan et al. (2002) 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 Nolan and Lee (2011)).

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 ($\mathrm{NN}_b(\cdot; \boldsymbol{\theta})$) is used to learn the right-hand-side (RHS) of the ODE:

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

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 (Krischer et al., 1992); 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 $\hat{\boldsymbol{v}}_I$. 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 $\boldsymbol{f}_{\text{kin}}$ 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
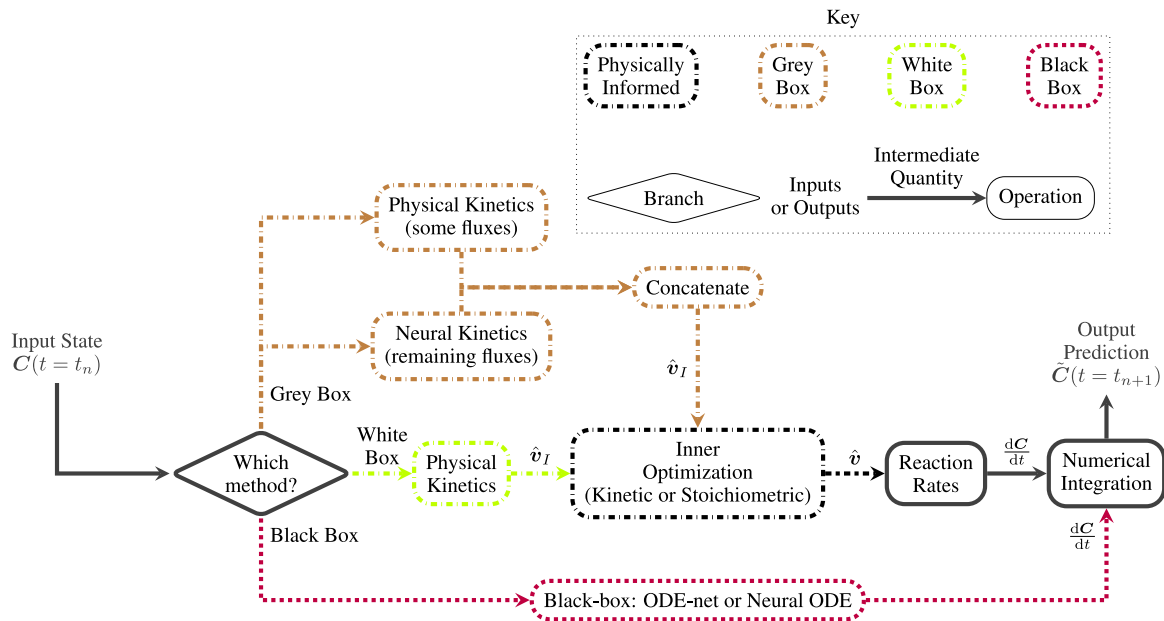
Fig. 1. White, gray, and black inner architectures. Operations are boxed, data or predictions are unboxed, and notable named intermediates are labeled on edges Color and pattern are used to distinguish between model pathways that are distinct between the gray-, white-, and black-box approaches, or common to all three.

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

constants	$\mathbf{S}_I, \mathbf{S}_E,$ and $\mathbf{B}_{ir}$;	and
outputs from		(11)
upstream modules	$\hat{\boldsymbol{v}}_I$	(function evaluations).

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 $\boldsymbol{\alpha}$ 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):

$$\boldsymbol{\alpha} \quad \text{and} \quad \theta. \tag{12}$$

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

the fluxes	$\boldsymbol{v}_E$ and $\boldsymbol{v}_I$;	and, possibly
auxiliary variables (described below)	$\boldsymbol{r}$.	

$$\tag{13}$$

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 (Amos and Kolter, 2017). This package itself uses `cvxpy` (a Python-embedded modeling language for convex optimization problems) (Diamond and Boyd, 2016; Agrawal et al., 2018, 2019a) and `diffcp` (a Python package for computing the derivative of a cone program, which is a special case of convex programming) (Agrawal et al., 2019c,b; Amos, 2019).

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 (Agrawal et al., 2019a) 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 $(\mathbf{S}_I \cdot \hat{\boldsymbol{v}}_I + \mathbf{S}_E \cdot \boldsymbol{v}_E)^T \cdot (\mathbf{S}_I \cdot \hat{\boldsymbol{v}}_I + \mathbf{S}_E \cdot \boldsymbol{v}_E)$ in the objective function of Eq. (5) has to be reformulated. This is resolved by the addition of another variable $\boldsymbol{r}$ in Eqs. (14) and (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 $\mathbf{S}_I \cdot \hat{v}_I$, we need to include v_I as an optimization variable, but then upgrade what was a pre-optimization expression $v_I = \hat{v}_I$ from Eq. (5) to an actual equality constraint in Eq. (14).

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

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

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

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

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 v_I and 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 Goyal et al. (2016)). 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

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

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

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

where $L+1$ is the length of the dataset $\{C(t = t_i) : i = 0, 1, 2, \ldots, L\}$ 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.

Table 2
Summaries of computational experiments in this paper.

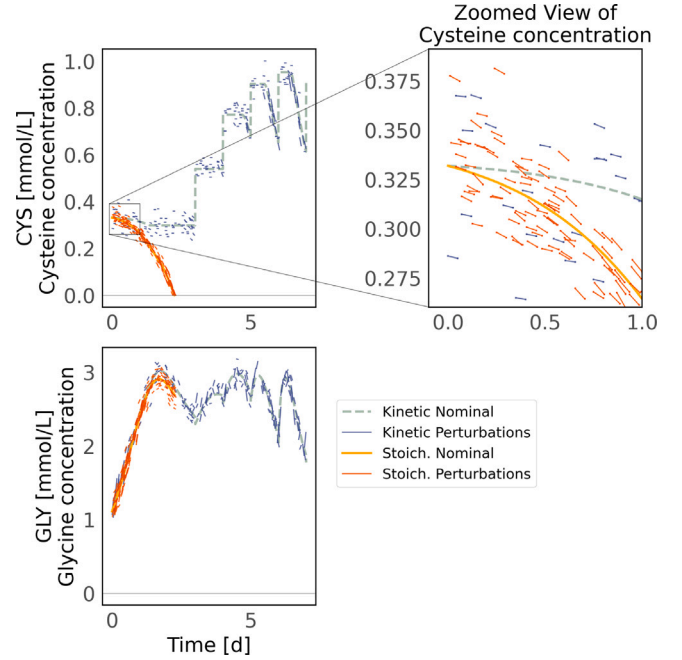| Experiments	Section
Data generation	Section 3.1
Black-box	Section 3.2
White-box (2 Parameters Unknown)	Section 3.3.1
White-box (5 Parameters Unknown)	Section 3.3.2
Gray-box (1 Expression Unknown)	Section 3.4.1
Gray-box (1 Expression Unknown + 1 Parameter Unknown)	Section 3.4.2

Fig. 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.

3.1. Data generation

We begin by simulating short trajectories for a variety of initial conditions, and collecting these flows as a dataset f or 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

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

where the nominal trajectory $\bar{C}(t) = (\bar{C}_1(t), \bar{C}_2(t), \ldots, \bar{C}_K(t))^T$ starts from a particular set of initial conditions that were measured during a laboratory experiment. The feeding events were implemented as state discontinuities. This nominal trajectory (in both its "kinetic" and its "stoichiometric" integrations) appears in Fig. 2; the "stoichiometric"

(a) Glycine; first derivative

(b) Glycine; second derivative

(c) Cysteine; first derivative

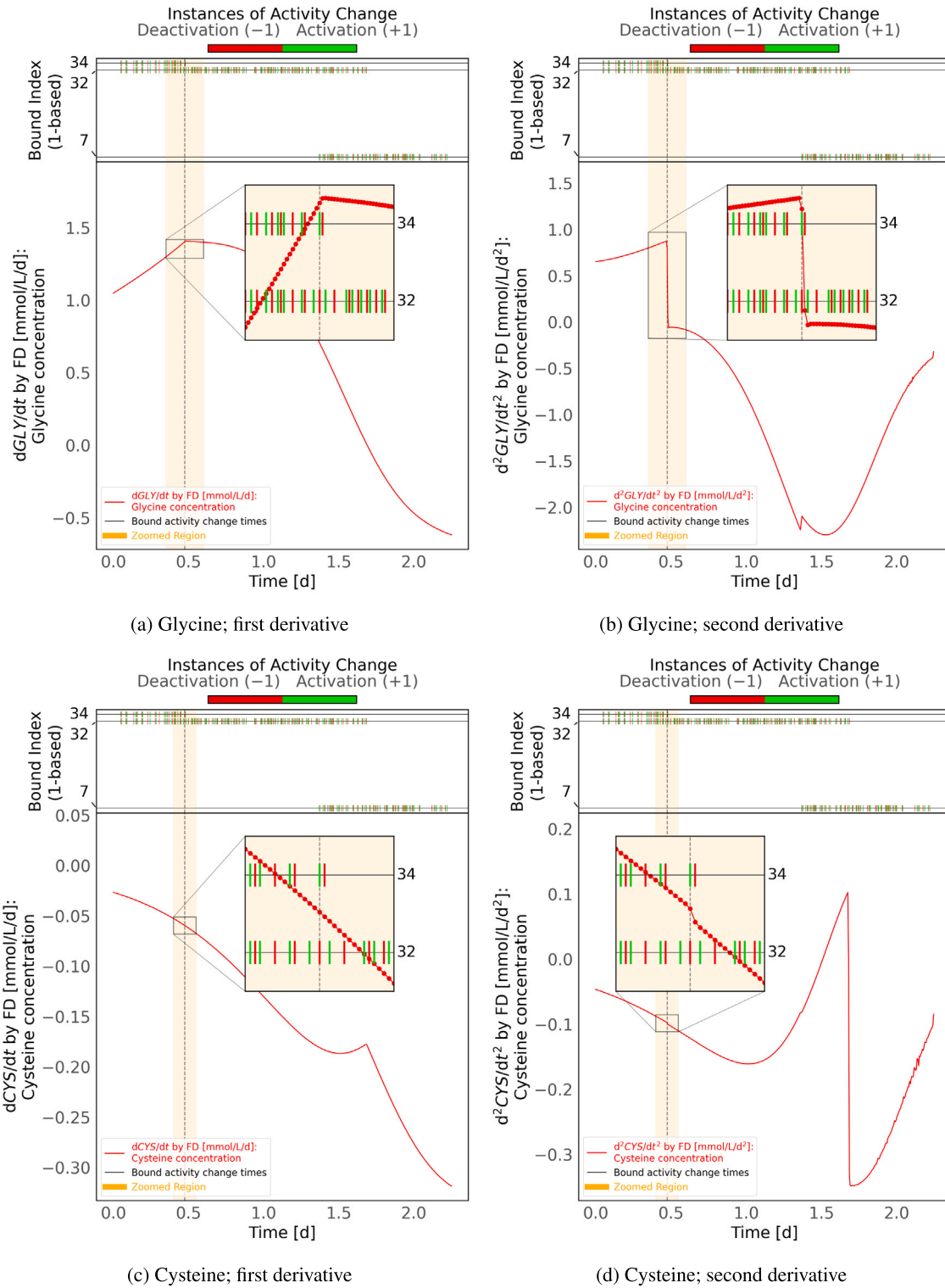(d) Cysteine; second derivative

Fig. 3. Activity of bound constraints along a sample run for the stoichiometric case, Observe the discontinuities arising in the second derivative (3(b) and 3(d)) 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.
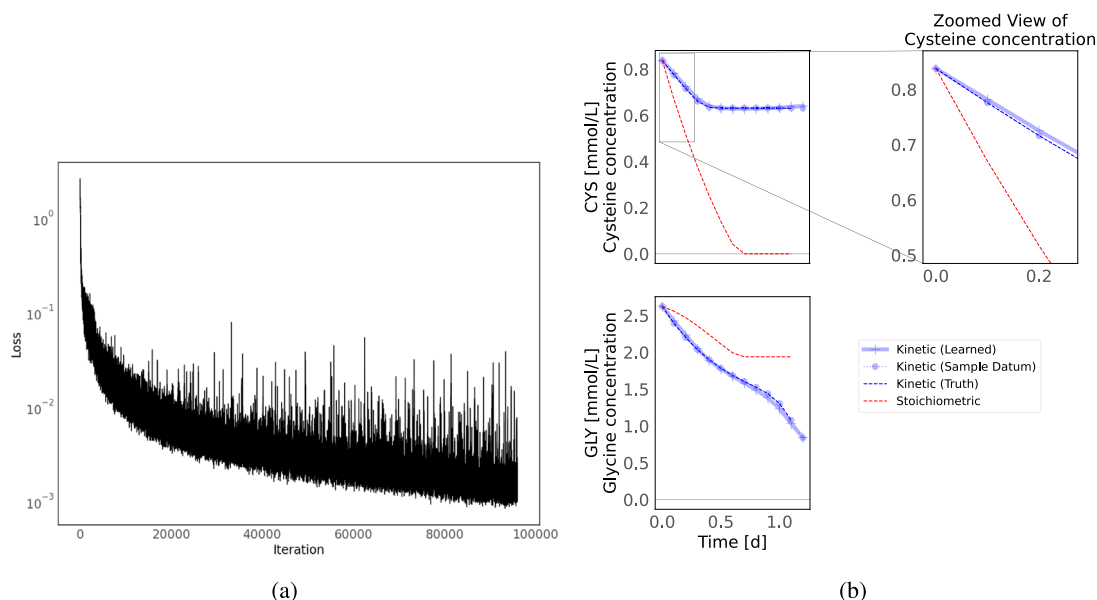
Fig. 4. Black-box training results (kinetic). See also Fig. 16 for more detailed results. Color is used to distinguish between different curves described in the legend.

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 (Hairer et al., 1993) 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 Figs. 4 and 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 Gomez et al. (2014), 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) (Gomez et al., 2014; Barton, 2000); 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 (3(c), 3(a)) and second (3(d), 3(b)) 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 Nolan and Lee (2011)); and that, at this time, its flux continuously changes from 0 to positive. We therefore expect, $\mathrm{d}v_{34}/\mathrm{d}t$ 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 $\mathrm{d}GLY/\mathrm{d}t \sim -v_{34}$, and $\mathrm{d}v_{34}/\mathrm{d}t$ is discontinuous, we can expect that $\mathrm{d}(\mathrm{d}GLY/\mathrm{d}t)/\mathrm{d}t$ will also be discontinuous, and that is clearly visible in Fig. 3(b).

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. 3(d)). 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 1.4); this is related to flux 7, which however involves different pathways.

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 $\Delta t = 0.1$ h over a $t_{\max} = 1.2$-h 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
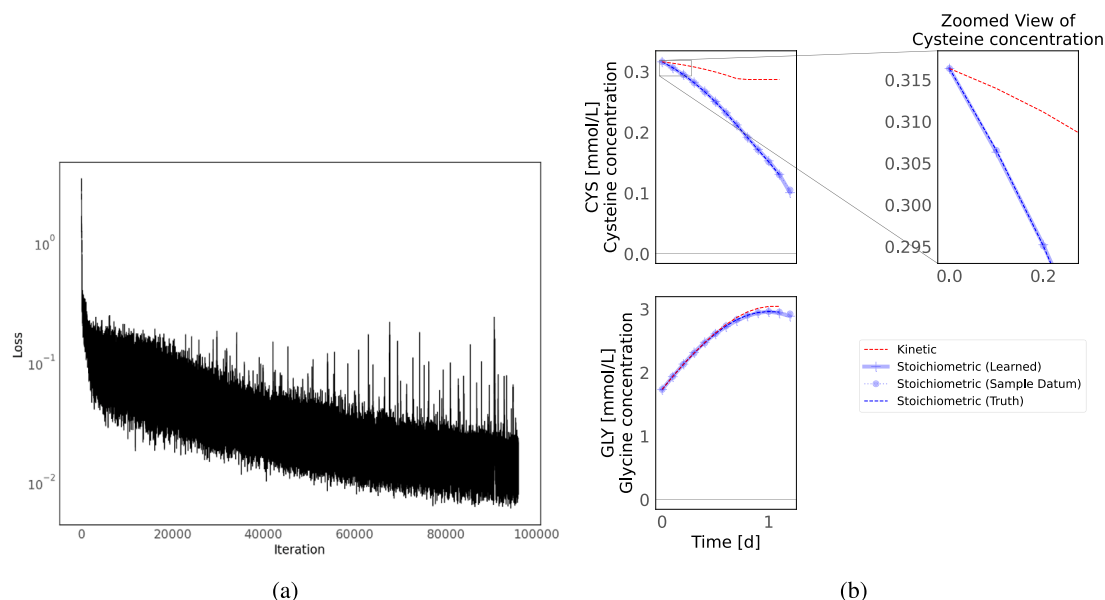
Fig. 5. Black-box training results (stoichiometric). See also Fig. 17 for more detailed results. Color is used to distinguish between different curves described in the legend.

black box identification "does not know" about discontinuities in the model - it smoothly interpolates between data points in time).

As can be seen in Figs. 4 and 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 "smoothened") closed-form functions.

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 h 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 Amos and Kolter (2017) 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 5; we find (Fig. 6) that we can recover the two parameter values reasonably well. For this approach, we chose the glucose utilization reaction, since glucose is one of the major nutrients and sources of energy for the cell; glucose feeds into many key reactions such as the TCA cycle and leads to lactate production. 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 (Kingma and Ba,
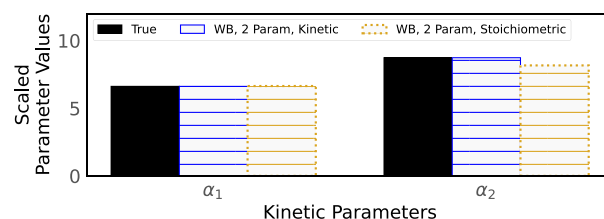

Fig. 6. Parameter comparison for white-box two-parameter.

2014) 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 (Figs. 7(b) and 8(b)), 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. 7(b), 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" (Daniels et al., 2008; Holiday et al., 2019); along the bottom of the trough the loss function posed is not strongly sensitive to the parameters, leading to parameter nonidentifiability.

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. For this approach we chose the lactate production reaction, since it is downstream from the glucose utilization reaction; it is also in itself an important reaction, because lactate is one the key by-products of cellular metabolism, known to inhibit key reactions such as antibody production. 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. An important point to note here is that, while other parameters selections have not been attempted for the two and five unknown parameter approaches, we expect that the results of parameter estimation would depend strongly on the model sensitivity to parameter perturbations. For example, given that biomass generation
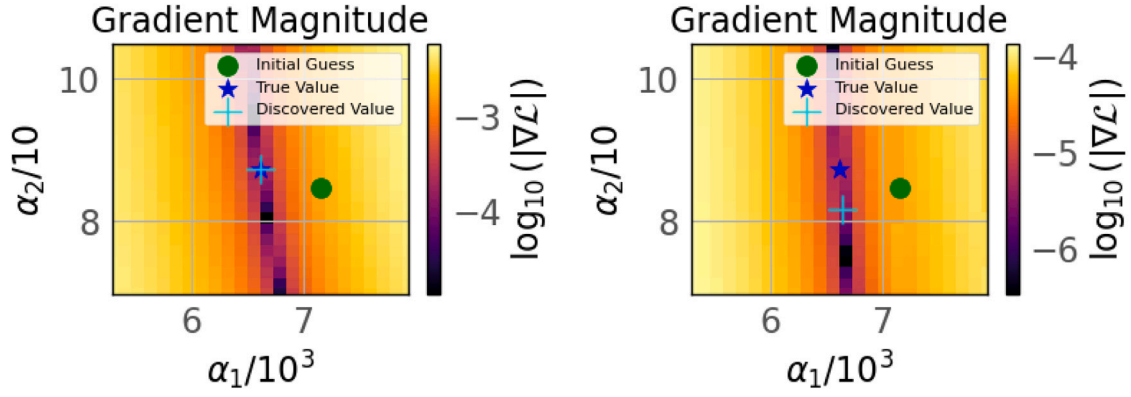
(a) **Kinetic (Type 1):** data is from ground-truth kinetic simulation; WB ansatz is also kinetic simulation.

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

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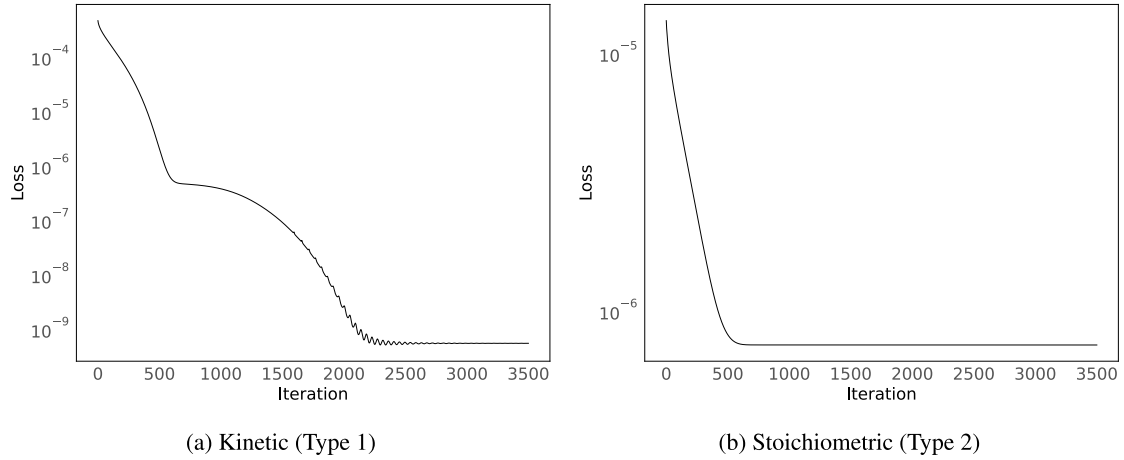

(a) Kinetic (Type 1)

(b) Stoichiometric (Type 2)

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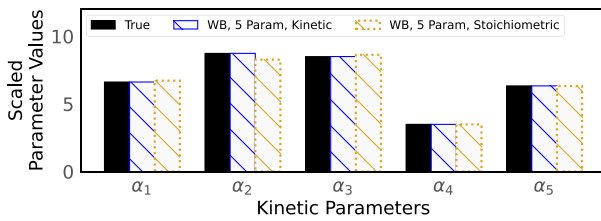

Fig. 9. Parameter comparison for the white-box, five-parameter case.

affects the VCD, which in turn affects the concentrations of all the dynamically tracked species, we might expect parameters directly associated with the biomass generation reaction to have a strong effect on prediction, and therefore be good candidates for future identification.

3.4. Partially known model: Gray-box identification

For the work in this section, we assumed the expression of $\hat{v}_{I,2}$ in $f_{\mathbf{kin}}$ 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 h 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 and formulation. Here again, as with the white box experiment, we decided to focus on the lactate production/utilization reaction, which affects other metabolic reactions as well; it represents an important metabolic process in the cell, and lactate is a crucial by-product of cellular metabolism. We used our neural-network-identified expression instead of the ground-truth to compute the flux value $\hat{v}_{I,2}$ to be fed into the inner optimization. Moreover, for the partially known parameters case, we chose a parameter from the glucose utilization reaction, because it directly feeds into the lactate generation reaction, and glucose is one of the key nutrients for the cell.

For each experiment, we find (Figs. 11(a) and 12(a), resp.) that the learned flux functions are largely reproduced correctly, but there are

(a) Kinetic (Type 1)   (b) Stoichiometric (Type 2)
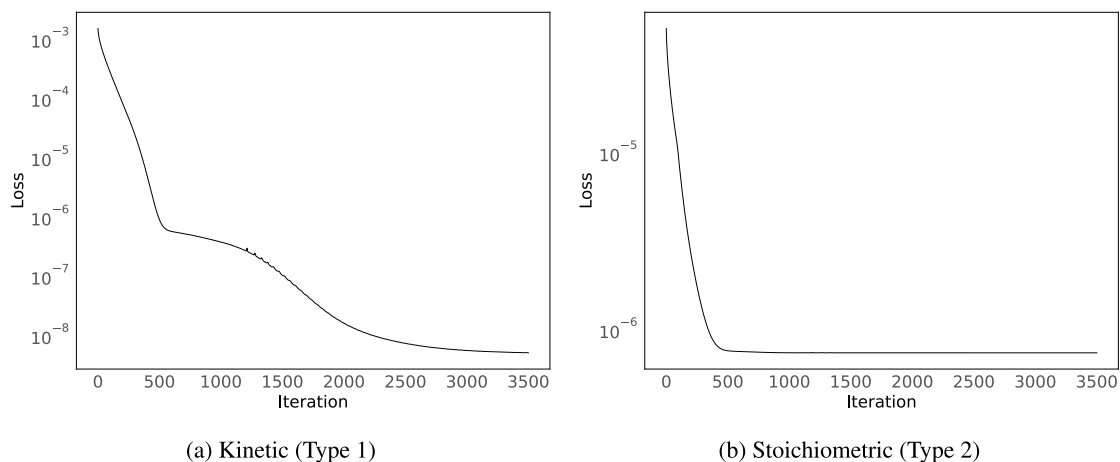
Fig. 10. Convergence of the training to the final parameter estimates for the white-box, five-parameter case. Kinetic (10(a)) vs. stoichiometric (10(b)) implementation from Section 2.1 See also Fig. 19.

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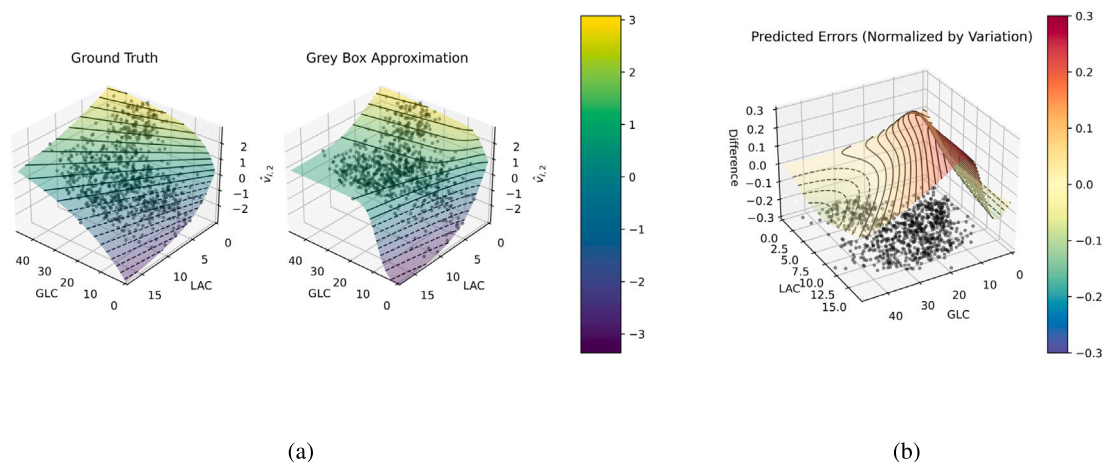
Fig. 11. Comparison of fluxes (ground-truth vs. kinetic gray-box model). Note that the (GLC, LAC) points visited in training are scattered on the surfaces (11(a)) or on the base plane (11(b)) 11(a): ground-truth and neural net approximations of the fluxes given the inputs of the neural net (GLC and LAC)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

(a)   (b)

Fig. 12. Comparison of fluxes (ground-truth vs. stoichiometric gray-box model). 12(a): ground-truth and neural net approximations of the fluxes given the neural net inputs, GLC and LAC. 12(b): 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 (12(a)) or on the base plane (12(b)). Note again the relative rotation (for visual clarity) between (12(a)) and (12(b)).

discrepancies (relatively flat network predictions) over some parts of their domain. The greatest percent discrepancy (given in Figs. 11(b)

and 12(b), scaled by the spread of true values across the training data) arises, as one might expect, at locations where the flux is approximately

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

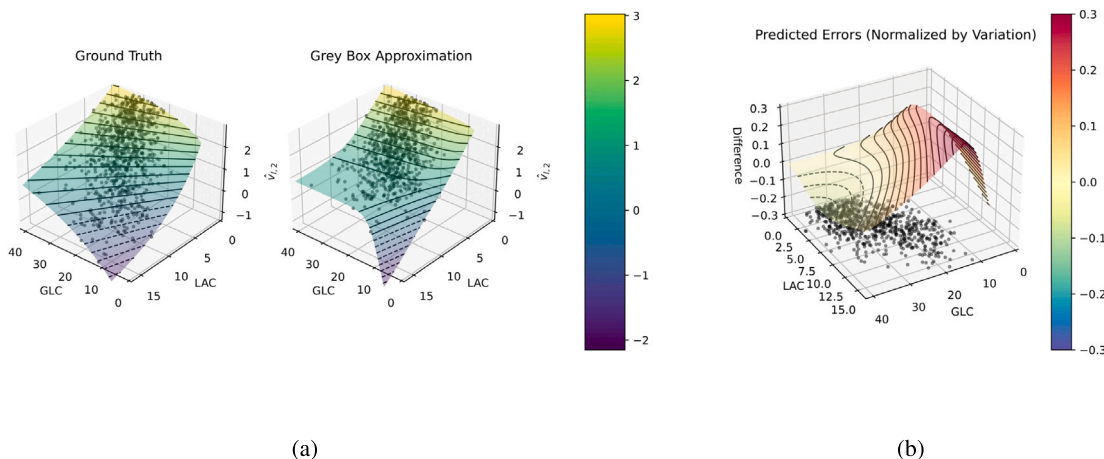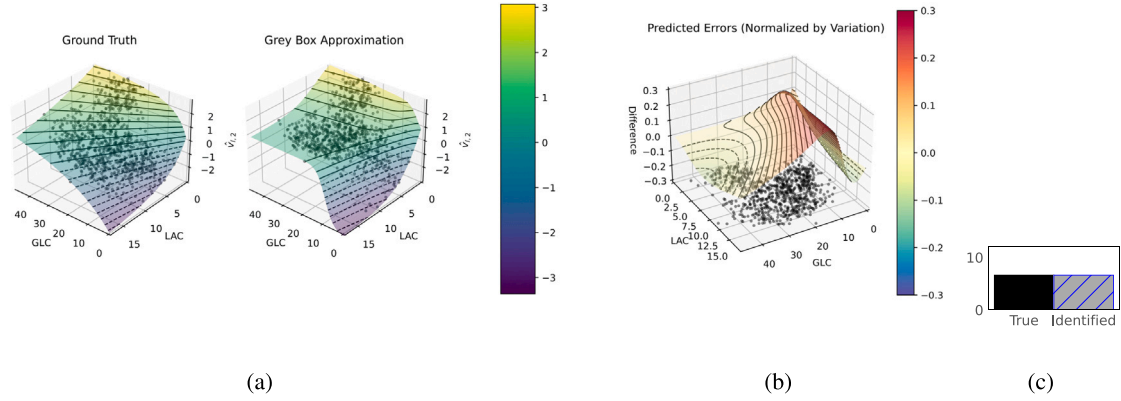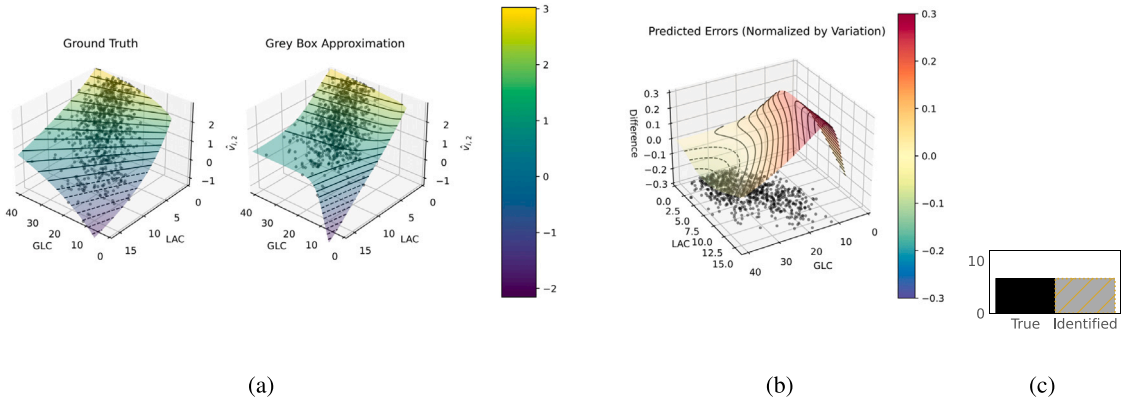**Fig. 14.** Comparison of fluxes (stoichiometric ground-truth vs. gray-box model). 14(a): ground-truth and neural net approximations of the fluxes as functions of the learned kinetic expression inputs GLC and LAC 14(b): normalized (as in Figs. 11(b) and 12(b)) predicted errors Black data points plotted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

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.

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 Figs. 13, 14 and Table 5 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.

In Table 3, we show some evaluations of right-hand-side timings for the ODEs evaluated using the alternate paths through Fig. 1. While further analysis is out of the scope of this paper, we can see immediately

Table 3
Comparison of compute times statistics through the Black box, White box and Gray box paths in Fig. 1. Times are wall-clock evaluations taken after training. Statistics are evaluated over about 800 realistic state vectors sampled from the same data used for training.

Model	Minimum (ms)	Average (ms)
White box	34.8	42.2
Black box	0.115	0.119
Gray box	12.4	20.3

that black box methods are two orders of magnitude faster than those relying on an internal optimization step; though all three are still much faster than real-time.

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

Table 4

Names and units for all variables $C = (C_1, C_2, \ldots, C_{14})^T$.

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 5

Names and ground-truth values for all parameters $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{45})^T$, 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.

	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
Section 3.3.1	Values (WB 2P Kin)	6617.2	87.354							
	Values (WB 2P Sto)	6643.8	81.666							
Section 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				
Section 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/L	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

"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 Amos and Kolter (2017): 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 an 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 (Amos et al., 2018) 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. Moreover, we can incorporate the CHO cell model developed in this paper into a broader bioreactor model for estimation of macroscopic trends, including eventually the effects of spatial inhomogeneity that may arise due to improper mixing due to scaling up (Lara et al., 2006; Ozturk, 1996), as well as the effects of population heterogeneity. Indeed, we have published on the behavior of heterogeneous and asynchronous agents in a reactor system (Thiem et al., 2021; Psarellis et al., 2023; Choi et al., 2016) while other authors have presented this concept specifically for cell culture bioreactors but not for dynamic FBA models such as ours (Karra et al., 2010; Bayrak et al., 2015).

CRediT authorship contribution statement

Table 6

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 "↔" are reversible, while "→" 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 4.

ID	Kin.	Stoi.	Kin. - Stoi.	Reaction
1	1.18	2.22	−1.04	G6P → 2PYR + 3ATP + 2NADH(Cytosolic)
2	0.0801	−0.901	0.981	PYR + NADH(Cytosolic) ↔ LAC
3	30.9	28.6	2.32	PYR + GLU ↔ ALA + AKG
4	1.024×10^{-12}	1.153×10^{-15}	1.023×10^{-12}	PYR + OXA → AKG + 2CO$_2$ + 2NADH(Mitochondrial)
5	22.5	20.4	2.06	AKG → MAL + CO$_2$ + NADH(Mitochondrial) + FADH$_2$ + ATP
6	1.151×10^{-13}	7.718×10^{-16}	1.144×10^{-13}	MAL ↔ OXA + NADH(Mitochondrial)
7	25.7	22.3	3.35	MAL → PYR + CO$_2$
8	1.36	1.36	$−2.852 \times 10^{-5}$	GLN ↔ GLU + NH$_3$
9	0.354	3.78	−3.42	AKG + NH$_3$ + NADH(Mitochondrial) ↔ GLU
10	5.23	5.23	$−3.530 \times 10^{-5}$	ASN ↔ ASP + NH$_3$
11	1.21	1.82	−0.61	ASP + AKG ↔ OXA + GLU
12	0.332	−0.597	0.929	SER + CO$_2$ + NH$_3$ + NADH(Cytosolic) ↔ 2GLY
13	4.34	3.41	0.929	Cystine + NADH(Cytosolic) → 2Cysteine
14	14.8	19.4	−4.55	NADH(Mitochondrial) + 0.5O$_2$ → 2.5ATP
15	16.6	20.5	−3.89	FADH$_2$ + 0.5O$_2$ → 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 → BIOM + 0.004FADH$_2$+0.0082GLU + 0.4445MAL + 0.6391NADH(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 → ANTI
18	3.17	4.27	−1.1	BIOM → BIOM(Extracellular)
19	2.68	7.34	−4.66	ANTI → ANTI(Extracellular)
20	4.28	2.89	1.39	GLC(Extracellular) + ATP → G6P
21	0.0801	−0.901	0.981	LAC ↔ LAC(Extracellular)
22	30.3	27.8	2.56	ALA ↔ ALA(Extracellular)
23	5.54	5.65	−0.115	ASN(Extracellular) → ASN
24	3.62	2.78	0.846	ASP ↔ ASP(Extracellular)
25	5.756×10^{-15}	3.41	−3.41	Cystine(Extracellular) + GLU → Cystine + GLU(Extracellular)
26	1.8	2.06	−0.255	GLN(Extracellular) ↔ GLN
27	0.0431	−1.96	2.01	GLY ↔ GLY(Extracellular)
28	0.249	3.07	−2.82	SER(Extracellular) → SER
29	5.9	3.4	2.5	NH$_3$ ↔ NH$_3$(Extracellular)
30	19.9	23.2	−3.26	O$_2$(Extracellular) ↔ O$_2$
31	47.8	43.4	4.47	CO$_2$ → CO$_2$(Extracellular)
32	4.23	3.27	0.965	2Cysteine + O$_2$ → 2Cystine
33	8.539×10^{-15}	−22.8	22.8	GLU → GLU(Extracellular)
34	−5.6	4.368×10^{-15}	−5.6	NADH(Cytosolic) → 0.5NADH(Mitochondrial) + 0.5FADH$_2$
35	0.483	−1.26	1.75	G6P + ATP + 2GLU → 2NADH(Cytosolic) + 2SER + 2AKG

review & editing. **Ioannis G. Kevrekidis:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors of hereby declare no conflicts of interest regarding the content presented in this paper.

Data availability

Data will be made available on request.

Appendix A. Physical variables and kinetic parameters

See Tables 4 and 5.

Appendix B. The ODE expressions $\frac{\mathrm{d}C}{\mathrm{d}t} = f_{\mathbf{eqn:ode}}(C; v(C; \alpha))$

$$\frac{\mathrm{d}C_1}{\mathrm{d}t} = v_{18}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_2}{\mathrm{d}t} = v_{19}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_3}{\mathrm{d}t} = -v_{20}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_4}{\mathrm{d}t} = v_{21}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_5}{\mathrm{d}t} = v_{22}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_6}{\mathrm{d}t} = -v_{23}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_7}{\mathrm{d}t} = v_{24}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_8}{\mathrm{d}t} = -v_{25}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_9}{\mathrm{d}t} = -v_{26}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_{10}}{\mathrm{d}t} = v_{27}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_{11}}{\mathrm{d}t} = -v_{28}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_{12}}{\mathrm{d}t} = v_{29}\frac{C_{14}}{1000}$$

$$\frac{\mathrm{d}C_{13}}{\mathrm{d}t} = (v_{25} + v_{33})\frac{C_{14}}{1000}$$

$$\hat{v}_{I,1} = \frac{v_{max1}\frac{\text{GLC}}{K_{m1}}}{1 + \frac{\text{LAC}}{K_{i1}} + \frac{\text{GLC}}{K_{m1}} + \frac{\text{LAC}}{K_{i1}}\frac{\text{GLC}}{K_{m1}}}$$

$$\hat{v}_{I,2} = \frac{v_{max2f}\frac{\text{GLC}}{K_{m2glc}} - v_{max2r}\frac{\text{LAC}}{K_{m2lac}}}{1 + \frac{\text{GLC}}{K_{m2glc}} + \frac{\text{LAC}}{K_{m2lac}}}$$

$$\hat{v}_{I,3} = \frac{v_{max3f}\frac{\text{GLC}}{K_{m3glc}} - v_{max3r}\frac{\text{ALA}}{K_{m3ala}}}{1 + \frac{\text{GLC}}{K_{m3glc}} + \frac{\text{ALA}}{K_{m3ala}}}$$

$$\hat{v}_{I,4} = \frac{v_{max8f}\frac{\text{GLN}}{K_{m8gln}} - v_{max8r}\frac{\text{GLU}}{K_{m8glu}}\frac{\text{NH3}}{K_{m8nh3}}}{1 + \frac{\text{GLN}}{K_{m8gln}} + \frac{\text{GLU}}{K_{m8glu}} + \frac{\text{NH3}}{K_{m8nh3}} + \frac{\text{GLU}}{K_{m8glu}}\frac{\text{NH3}}{K_{m8nh3}}}$$

$$\hat{v}_{I,5} = \frac{v_{max9f}\frac{\text{NH3}}{K_{m9nh3}} - v_{max9r}(\frac{\text{GLU}}{K_{m9glu}} + \frac{\text{GLN}}{K_{m9gln}})}{1 + \frac{\text{NH3}}{K_{m9nh3}} + \frac{\text{GLU}}{K_{m9glu}} + \frac{\text{GLN}}{K_{m9gln}} + \frac{\text{GLU}}{K_{m9glu}}\frac{\text{GLN}}{K_{m9gln}}}$$

$$\hat{v}_{I,6} = \frac{v_{max10f}\frac{\text{ASN}}{K_{m10asn}} - v_{max10r}\frac{\text{ASP}}{K_{m10asp}}\frac{\text{NH3}}{K_{m10nh3}}}{1 + \frac{\text{ASN}}{K_{m10asn}} + \frac{\text{ASP}}{K_{m10asp}} + \frac{\text{NH3}}{K_{m10nh3}} + \frac{\text{ASP}}{K_{m10asp}}\frac{\text{NH3}}{K_{m10nh3}}}$$

$$\hat{v}_{I,7} = \frac{v_{max11}(\frac{\text{ASP}}{K_{m11asp}} + \frac{\text{ASN}}{K_{m11asn}})}{1 + \frac{\text{ASN}}{K_{m11asn}} + \frac{\text{ASP}}{K_{m11asp}} + \frac{\text{ASP}}{K_{m11asp}}\frac{\text{ASN}}{K_{m11asn}}}$$

$$\hat{v}_{I,8} = \frac{v_{max12f} \cdot \hat{v}_{I,10}\frac{\text{SER}}{K_{m12ser}}\frac{\text{NH3}}{K_{m12nh3}} - v_{max12r}(\frac{\text{GLY}}{K_{m12gly}})^2}{1 + \frac{\text{SER}}{K_{m12ser}} + \frac{\text{GLY}}{K_{m12gly}} + \frac{\text{NH3}}{K_{m12nh3}} + (\frac{\text{GLY}}{K_{m12gly}})^2}$$

$$\hat{v}_{I,9} = \frac{v_{max13}\frac{\text{CYS}}{K_{m13}}}{1 + \frac{\text{CYS}}{K_{m13}}}$$

$$\hat{v}_{I,10} = \frac{v_{max16}\frac{\text{GLN}}{K_{m16a}}\frac{\text{ASN}}{K_{m16b}}\frac{\text{ALA}}{K_{m16c}}}{1 + \frac{\text{GLN}}{K_{m16a}} + \frac{\text{ASN}}{K_{m16b}} + \frac{\text{ALA}}{K_{m16c}} + \frac{\text{GLN}}{K_{m16a}}\frac{\text{ASN}}{K_{m16b}} + \frac{\text{GLN}}{K_{m16a}}\frac{\text{ALA}}{K_{m16c}} + \frac{\text{ASN}}{K_{m16b}}\frac{\text{ALA}}{K_{m16c}} + \frac{\text{GLN}}{K_{m16a}}\frac{\text{ASN}}{K_{m16b}}\frac{\text{ALA}}{K_{m16c}}}$$

$$\hat{v}_{I,11} = \frac{v_{max17}}{1 + \frac{\text{LAC}}{K_{i17}}}$$

$$\hat{v}_{I,12} = \hat{v}_{I,10}$$

$$\hat{v}_{I,13} = \hat{v}_{I,11}$$

$$\hat{v}_{I,14} = \frac{v_{max35}\frac{\text{GLC}}{K_{m35a}}(\frac{\text{GLU}}{K_{m35b}})^2}{1 + \frac{\text{GLC}}{K_{m35a}} + \frac{\text{GLU}}{K_{m35b}} + (\frac{\text{GLU}}{K_{m35b}})^2}$$

Box I.

$$\frac{dC_{14}}{dt} = \frac{\frac{dC_1}{dt}}{2.31} - 0.003C_{14}$$

Appendix C. List of all reactions

See Table 6.

Appendix D. The kinetic expressions $\hat{v}_I = f_{\mathbf{kin}}(C; \alpha)$

See Box I.

Appendix E. Deviations in kinetic expressions from Nolan and Lee (2011)

All kinetically defined intracellular reactions have been modeled using the Michaelis Menten format, based on kinetic expressions from Nolan and Lee (Nolan and Lee, 2011). 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 [1]3C-labeled (intracellular and extracellular) metabolite tracking data. Reasons for the modifications are listed below:

- Reaction 1 (corresponding to $\hat{v}_{I,1}$): G6P → 2PYR + 3ATP+2NADH Temperature dependent constant TC_1 and inhibition exponential constant exp_1 (which is also temperature dependent) were removed from the kinetic expression.
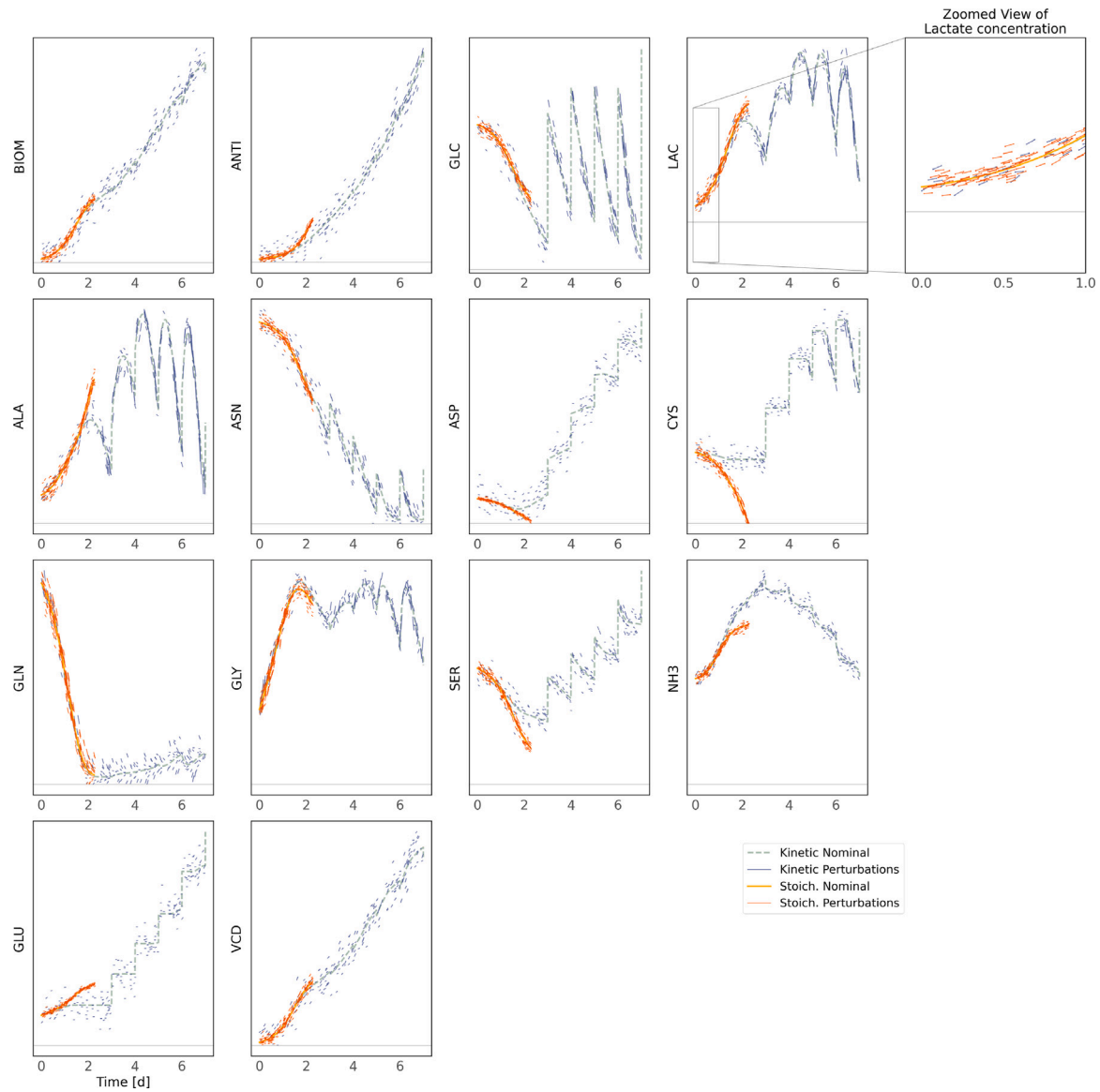
Fig. 15. Nominal trajectories overlaid on sampled short-time flows from perturbed initial conditions. Color is used to distinguish between different curves described in the legend.

- Reaction 2 (corresponding to $\hat{v}_{I,2}$): PYR + NADH $\leftrightarrow$ LAC
  Redox variable R removed and reaction made continuous.
- Reaction 3 (corresponding to $\hat{v}_{I,3}$): PYR + GLU $\leftrightarrow$ ALA + AKG
  Temperature constant TC_{3b} removed.
- Reaction 8 (corresponding to $\hat{v}_{I,4}$): GLN $\leftrightarrow$ GLU + NH$_3$
  Temperature constant TC_8 removed.
- Reaction 9 (corresponding to $\hat{v}_{I,5}$): AKG + NH$_3$ + NADH $\leftrightarrow$ GLU
  Dependence on intracellular metabolite AKG (not dynamically tracked) removed, to make reaction expression adhere to Michaelis Menten kinetics.
- Reaction 10 (corresponding to $\hat{v}_{I,6}$): ASN $\leftrightarrow$ ASP + NH$_3$
  Temperature dependent constant TC_{10} removed from kinetic expression.
- Reaction 11 (corresponding to $\hat{v}_{I,7}$): ASP + AKG $\leftrightarrow$ OXA + GLU
  The reaction expression has been changed to be irreversible.
- Reaction 12 (corresponding to $\hat{v}_{I,8}$): SER + CO$_2$ + NH$_3$ + NADH $\leftrightarrow$ 2GLY
  Forward reaction made dependent on ammonia in addition to serine.
- Reaction 13 (corresponding to $\hat{v}_{I,9}$): Cystine + NADH $\rightarrow$ 2Cysteine
  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 $\hat{v}_{I,14}$): G6P + ATP + 2GLU $\rightarrow$ 2NADH(Cytosolic) + 2SER + 2AKG
  This reaction was added based on the presence of the serine synthesis pathway (SSP) originating from glycolysis.

Appendix F. The stoichiometric matrix S

Blue: indices of intracellular fluxes

Olive Overlined: indices of reversible extracellular fluxes

Red Underlined: indices of irreversible extracellular fluxes

See Box II.

	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	−1	0	0	0	−1	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	2
ALA	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	−0.0838	−0.0614	0	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	0	1	0	−1	0	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	−0.041	−0.0344	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ASP	0	0	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	0	1	0	0	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	−1
BIOM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	−1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CO₂	0	0	0	2	1	0	1	0	0	0	0	−1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−1	0	0	0	0	0
Cysteine	0	0	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	−2	0	0	0	0	0
Cystine	0	0	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
FADH₂	0	0	0	0	1	0	0	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	0.5	0
G6P	−1	0	0	0	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	−1
GLU	0	0	−1	0	0	0	0	1	1	0	1	0	0	0	0	0.0082	−0.0479	0	0	0	0	0	0	−1	0	0	0	0	0	0	−1	0	0	0	−2
GLN	0	0	0	0	0	0	0	−1	0	0	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
GLY	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	−0.056	−0.0719	0	0	0	0	0	0	0	0	0	0	0	−1	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	0	−1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MAL	0	0	0	0	1	−1	−1	0	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
NADH (Cytosolic)	2	−1	0	0	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	−1	2
NADH (Mitochondrial)	0	0	0	2	1	1	0	0	−1	0	0	0	0	0	−1	0.6391	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0
NH₃	0	0	0	0	0	0	0	1	−1	1	0	−1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−1	0	0	0	0	0
O₂	0	0	0	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	0	0	0	1	0	−1	0	0
OXA	0	0	0	−1	0	1	0	0	0	1	0	0	0	0	0	−0.427	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PYR	2	−1	−1	−1	0	0	1	0	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
SER	0	0	0	0	0	0	0	0	0	0	0	−1	0	0	0	−0.096	−0.1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2

Box II.

Zoomed View of Lactate concentration

Panels (rows/columns): BIOM, ANTI, GLC, LAC; ALA, ASN, ASP, CYS; GLN, GLY, SER, NH3; GLU, VCD. x-axis: Time [d].

Legend:
— Kinetic (Learned)
⋯ Kinetic (Sample Datum)
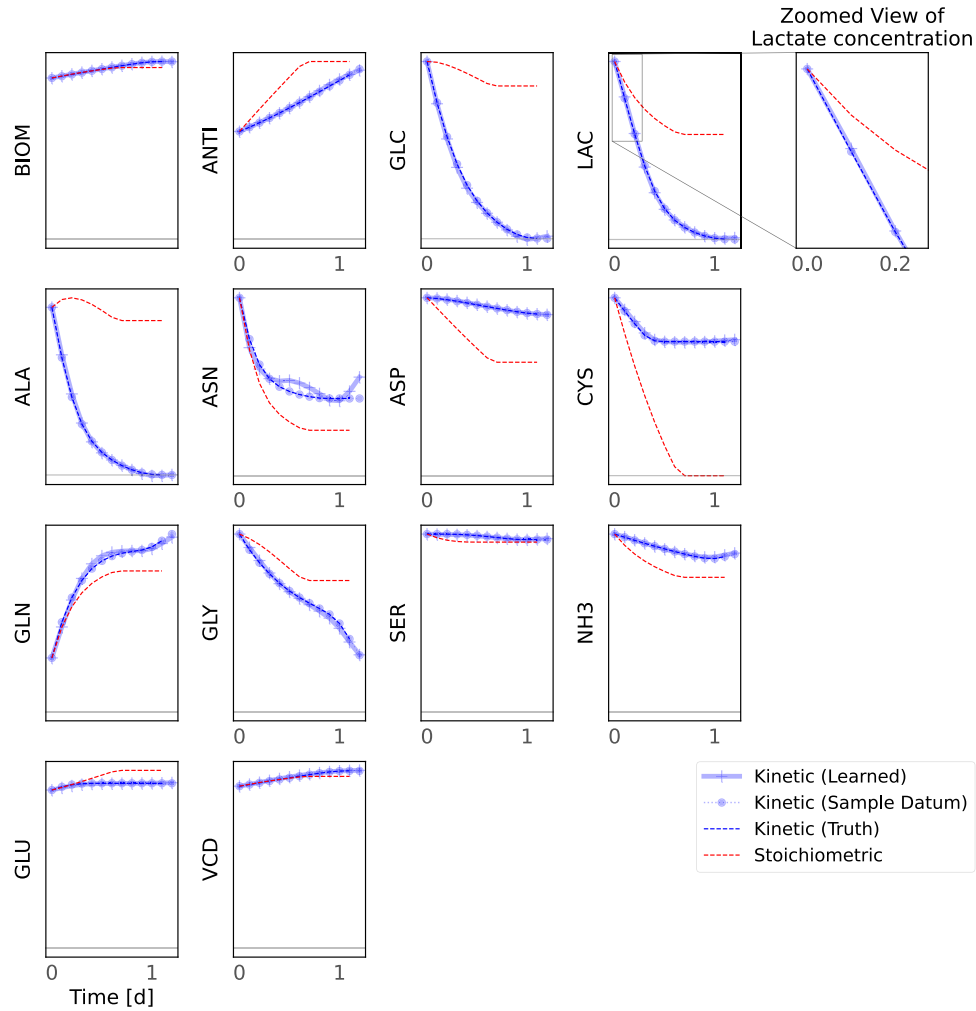-- Kinetic (Truth)
-- Stoichiometric

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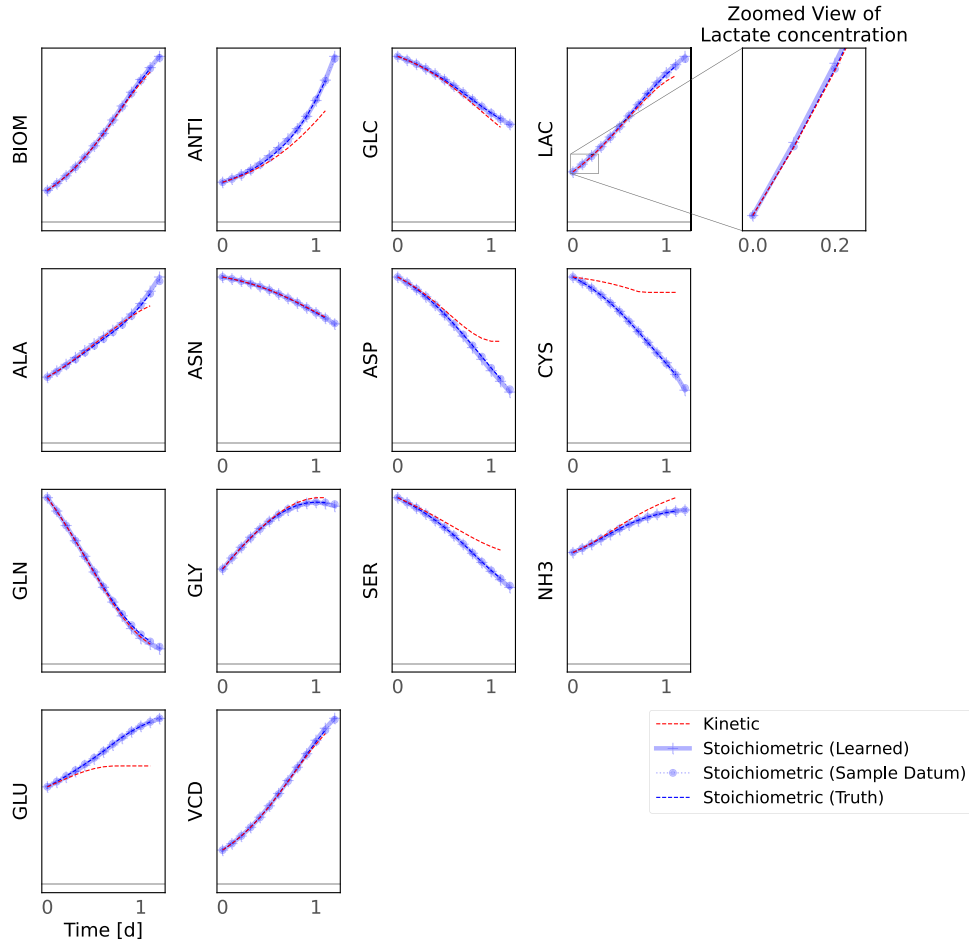
Fig. 17. Black-box predictions; all variables (Stoichiometric (Type 2)). Color is used to distinguish between different curves described in the legend.

Appendix G. Closed-form solution for stoichiometric-based approach (w/o inequality bounds)

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

$$\|\boldsymbol{v}_I - \hat{\boldsymbol{v}}_I\|_2^2 = \boldsymbol{v}_I^T \cdot \boldsymbol{v}_I - 2\hat{\boldsymbol{v}}_I \cdot \boldsymbol{v}_I + \hat{\boldsymbol{v}}_I^T \cdot \hat{\boldsymbol{v}}_I. \tag{19}$$

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

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

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

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

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

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

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

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

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

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

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

equations and variables:

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

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

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

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

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

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

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

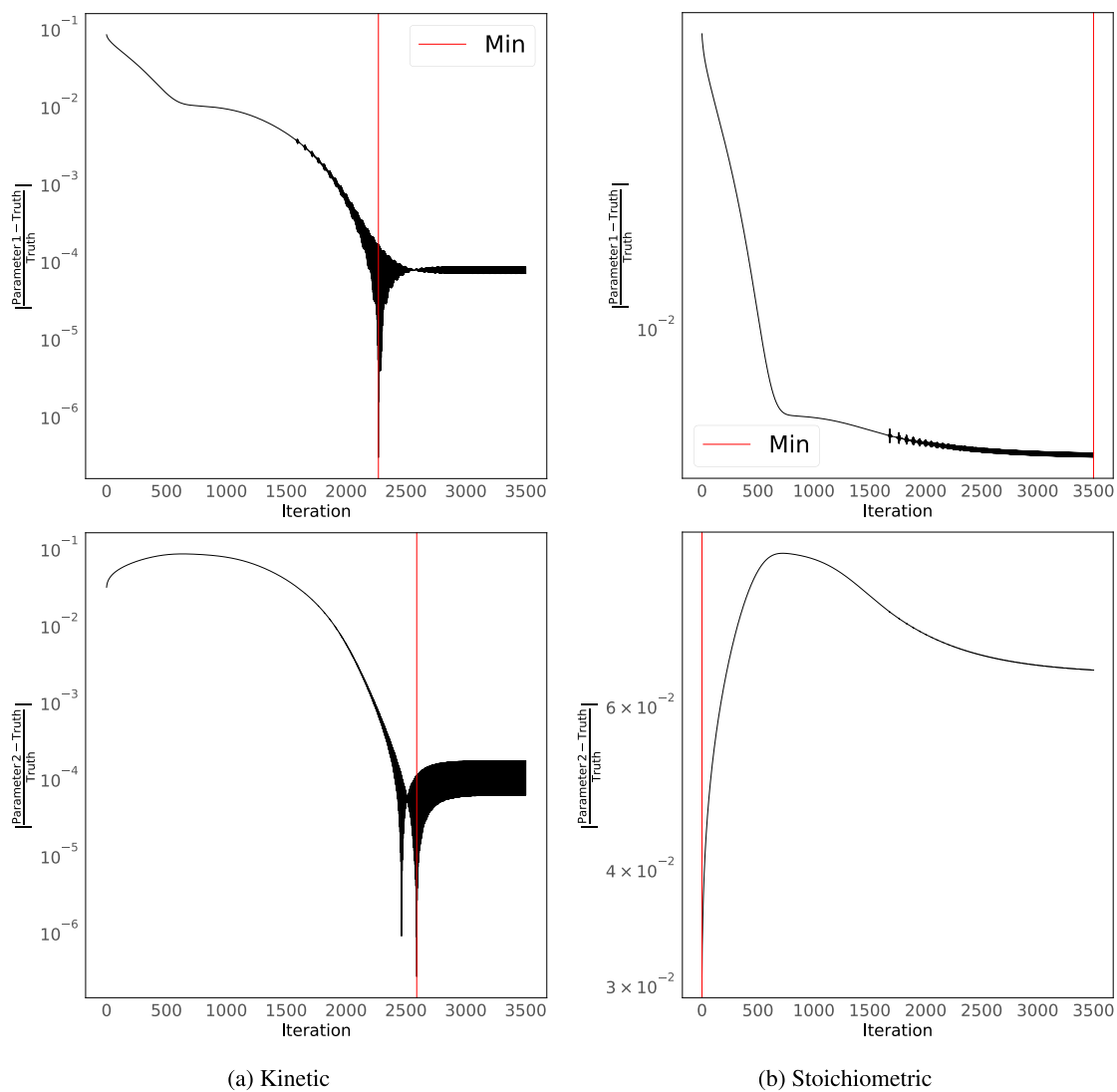is the closed-form expression for stoichiometric-based approach (w/o bounds).

(a) Kinetic

(b) Stoichiometric

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

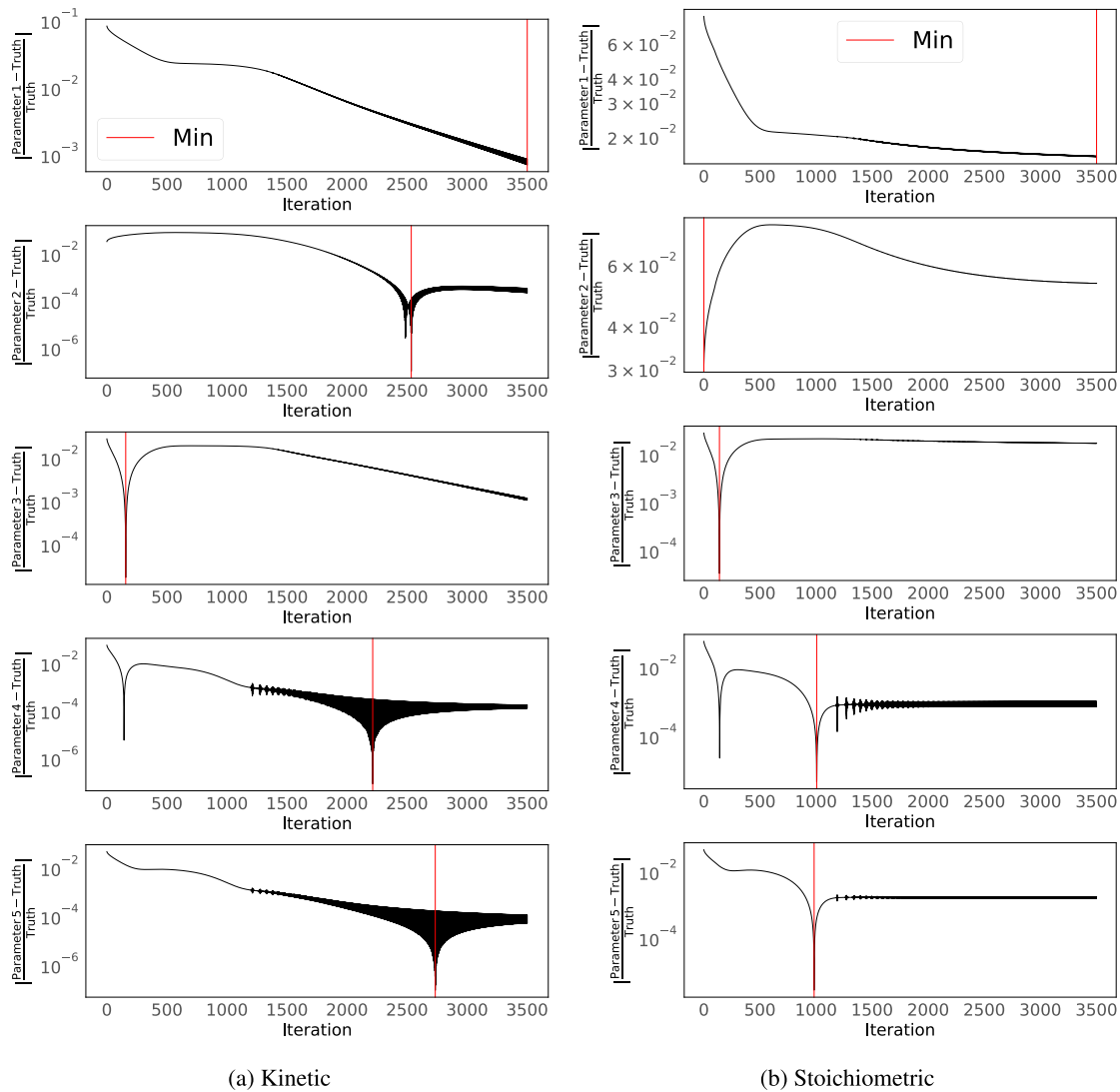(a) Kinetic   (b) Stoichiometric

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

Appendix H. Additional figures

See Figs. 15–19.

References

Adomaitis, R.A., Kevrekidis, I.G., Farber, R.M., Lapedes, A.S., Hudson, J.L., Kube, M., 1990. Application of Neural Nets to System Identification and Bifurcation Analysis of Real World Experimental Data. Technical report, Los Alamos National Lab, Los Alamos, NM (United States).

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., Kolter, J.Z., 2019a. Differentiable convex optimization layers. In: Advances in Neural Information Processing Systems. pp. 9558–9570.

Agrawal, A., Barratt, S., Boyd, S., Busseti, E., Moursi, W., 2019b. diffcp: differentiating through a cone program, version 1.0.19. https://github.com/cvxgrp/diffcp.

Agrawal, A., Barratt, S., Boyd, S., Busseti, E., Moursi, W., 2019c. Differentiating through a cone program. J. Appl. Numer. Optim. 1 (2), 107–115.

Agrawal, A., Verschueren, R., Diamond, S., Boyd, S., 2018. A rewriting system for convex optimization problems. J. Control Decis. 5 (1), 42–60.

Amos, Brandon, 2019. Differentiable Optimization-Based Modeling for Machine Learning (Ph.D. thesis). Carnegie Mellon University.

Amos, B., Jimenez, I., Sacks, J., Boots, B., Kolter, J.Z., 2018. Differentiable MPC for end-to-end planning and control. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), In: Advances in Neural Information Processing Systems, vol. 31, Curran Associates, Inc..

Amos, B., Kolter, J.Z., 2017. OptNet: Differentiable optimization as a layer in neural networks. In: International Conference on Machine Learning.

Barton, P.I., 2000. Modeling, simulation and sensitivity analysis of hybrid systems. In: CACSD. Conference Proceedings. IEEE International Symposium on Computer-Aided Control System Design (Cat. No.00TH8537). pp. 117–122.

Barton, P.I., Lee, C.K., 2002. Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. ACM Trans. Model. Comput. Simul. 12 (4), 256–289.

Bayrak, E.S., Wang, T., Cinar, A., Undey, C., 2015. Computational modeling of fed-batch cell culture bioreactor: Hybrid agent-based approach. IFAC-PapersOnLine 48 (8), 1252–1257.

Boghigian, B.A., Seth, G., Kiss, R., Pfeifer, B.A., 2010. Metabolic flux analysis and pharmaceutical production. Metab. Eng. 12 (2), 81–95.

Brunton, S.L., Kutz, J.N., 2022. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control, second ed. Cambridge University Press.

Butler, Michael, 2005. Animal cell cultures: recent achievements and perspectives in the production of biopharmaceuticals. Appl. Microbiol. Biotechnol. 68, 283–291.

Chassagnole, C., Noisommit-Rizzi, N., Schmid, J.W., Mauch, K., Reuss, M., 2002. Dynamic modeling of the central carbon metabolism of Escherichia coli. Biotechnol. Bioeng. 79 (1), 53–73.

Choi, M., Bertalan, T., Laing, C.R., Kevrekidis, I.G., 2016. Dimension reduction in heterogeneous neural networks: Generalized Polynomial Chaos (gPC) and ANalysis-Of-VAriance (ANOVA). Eur. Phys. J. Spec. Top. 225 (6–7), 1165–1180.

Daneker, M., Zhang, Z., Karniadakis, G.E., Lu, L., 2022. Systems Biology: Identifiability analysis and parameter identification via systems-biology informed neural networks. arXiv preprint arXiv:2202.01723.

Daniels, B.C., Chen, Y., Sethna, J.P., Gutenkunst, R.N., Myers, C.R., 2008. Sloppiness, robustness, and evolvability in systems biology. Curr. Opin. Biotechnol. 19 (4), 389–395, Protein technologies / Systems biology.

Diamond, S., Boyd, S., 2016. CVXPY: A Python-embedded modeling language for convex optimization. J. Mach. Learn. Res. 17 (83), 1–5.

Gomez, J.A., Höffner, K., Barton, P.I., 2014. DFBAlab: a fast and reliable MATLAB code for dynamic flux balance analysis. BMC Bioinformatics 15 (409).

Goudar, C., Biener, R., Boisart, C., Heidemann, R., Piret, J., de Graaf, A., Konstantinov, K., 2010. Metabolic flux analysis of CHO cells in perfusion culture by metabolite balancing and 2D [13C, 1H] COSY NMR spectroscopy. Metab. Eng. 12 (2), 138–149.

Goyal, A., Lamb, A., Zhang, Y., Zhang, S., Courville, A., Bengio, Y., 2016. Professor forcing: A new algorithm for training recurrent networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS '16, Curran Associates Inc., Red Hook, NY, USA, pp. 4608–4616.

Hairer, E., Nørsett, S.P., Wanner, G., 1993. Solving Ordinary Differential Equations I: Nonstiff Problems. Springer, Berlin, Heidelberg.

Holiday, A., Kooshkbaghi, M., Bello-Rivas, J.M., Gear, C.W., Zagaris, A., Kevrekidis, I.G., 2019. Manifold learning for parameter reduction. J. Comput. Phys. 392, 419–431.

Hudson, J.L., Kube, M., Adomaitis, R.A., Kevrekidis, I.G., Lapedes, A.S., Farber, R.M., 1990. Nonlinear signal processing and system identification: applications to time series from electrochemical reactions. Chem. Eng. Sci. 45 (8), 2075–2081.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat. Rev. Phys. 3 (6), 422–440.

Karra, S., Sager, B., Karim, M.N., 2010. Multi-scale modeling of heterogeneities in mammalian cell culture processes. Ind. Eng. Chem. Res. 49 (17), 7990–8006.

Kemeth, F.P., Alonso, S., Echebarria, B., Moldenhawer, T., Beta, C., Kevrekidis, I.G., 2022a. Black and gray box learning of amplitude equations: Application to phase field systems. arXiv e-prints, arXiv–2207.

Kemeth, F.P., Bertalan, T., Thiem, T., Dietrich, F., Moon, S.J., Laing, C.R., Kevrekidis, I.G., 2022b. Learning emergent partial differential equations in a learned emergent space. Nature Commun. 13 (1), 3318.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR, abs/1412.6980.

Krischer, K., Kevrekidis, I.G., Kube, M.C., Hudson, J.L., 1992. Discrete-vs. continuous-time nonlinear signal processing of Cu electrodissolution data. Chem. Eng. Commun. 118 (1), 25–48.

Kutz, J. Nathan, 2013. Data-Driven Modeling and Scientific Computation: Methods for Complex Systems and Big Data. Oxford University Press, Inc., USA.

Lara, A.R., Galindo, E., Ramírez, O.T., et al., 2006. Living with heterogeneities in bioreactors. Mol. Biotechnol. 34, 355–381.

Lee, S., Psarellis, Y.M., Siettos, C.I., Kevrekidis, I.G., 2022. Learning black-and gray-box chemotactic PDEs/closures from agent based Monte Carlo simulation data. arXiv e-prints, arXiv–2205.

Lu, L., Meng, X., Mao, Z., Karniadakis, G.E., 2021. DeepXDE: A deep learning library for solving differential equations. SIAM Rev. 63 (1), 208–228.

Ma, N., Ellet, J., Okediadi, C., Hermes, P., McCormick, E., Casnocha, S., 2009. A single nutrient feed supports both chemically defined NS0 and CHO fed-batch processes: Improved productivity and lactate metabolism. Biotechnol. Prog. 25 (5), 1353–1363.

Mahadevan, R., Edwards, J.S., Doyle, F.J., 2002. Dynamic flux balance analysis of diauxic growth in Escherichia coli. Biophys. J. 83 (3), 1331–1340.

Maranas, C.D., Zomorrodi, A.R., 2016. Optimization Methods in Metabolic Networks. John Wiley & Sons.

Martin-Linares, C.P., Psarellis, Y.M., Karapetsas, G., Koronaki, E.D., Kevrekidis, I.G., 2023. Physics-agnostic and Physics-infused machine learning for thin films flows: modeling, and predictions from small data. arXiv preprint arXiv:2301.12508.

Nolan, R.P., Lee, K., 2011. Dynamic model of CHO cell metabolism. Metab. Eng. 13 (1), 108–124.

Ozturk, S.S., 1996. Engineering challenges in high-density cell culture systems. Cytotechnology 22, 3–16.

Psarellis, Y.M., Kavousanakis, M., Henson, M.A., Kevrekidis, I.G., 2023. Limits of entrainment of circadian neuronal networks. Chaos 33 (1), 013137.

Psarellis, Y.M., Lee, S., Bhattacharjee, T., Datta, S.S., Bello-Rivas, J.M., Kevrekidis, I.G., 2022. Data-driven discovery of chemotactic migration of bacteria via machine learning. arXiv preprint arXiv:2208.11853.

Quek, L., Dietmair, S., Krömer, J.O., Nielsen, L.K., 2010. Metabolic flux analysis in mammalian cell culture. Metab. Eng. 12 (2), 161–171.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.

Rico-Martines, R., Kevrekidis, I.G., Kube, M.C., Hudson, J.L., 1993. Discrete-vs. continuous-time nonlinear signal processing: Attractors, transitions and parallel implementation issues. In: 1993 American Control Conference. IEEE, pp. 1475–1479.

Rico-Martinez, R., Anderson, J.S., Kevrekidis, I.G., 1994. 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. pp. 596–605.

Stephanopoulos, G.N., Aristidou, A.A., Nielsen, J., 1998. Metabolic Engineering: Principles and Methodologies. Academic Press.

Suthers, P.F., Foster, C.J., Sarkar, D., Wang, L., Maranas, C.D., 2021. Recent advances in constraint and machine learning-based metabolic modeling by leveraging stoichiometric balances, thermodynamic feasibility and kinetic law formalisms. Metab. Eng. 63, 13–33.

Thiem, T.N., Kemeth, F.P., Bertalan, T., Laing, C.R., Kevrekidis, I.G., 2021. Global and local reduced models for interacting, heterogeneous agents. Chaos 31 (7), 073139.

Yazdani, A., Lu, L., Raissi, M., Karniadakis, G.E., 2020. Systems biology informed deep learning for inferring parameters and hidden dynamics. PLoS Comput. Biol. 16 (11), 1–19.