Mathematical Biosciences 396 (2026) 109649

Contents lists available at ScienceDirect

Mathematical Biosciences

journal homepage: www.elsevier.com/locate/mbs

Original research article

Combining model-based and data-driven models: An application to synthetic biology resource competition

Atefe Darabi [a,1], Zheming An [b,c,d,1], Muhammad Ali Al-Radhawi [a,e], William Cho [e], Milad Siami [a], Eduardo D. Sontag [a,e,f,g,*]

[a] Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA
[b] Division of Genetics and Genomics, Department of Pediatrics, Boston Children's Hospital, Boston, MA, USA
[c] Harvard Medical School, Boston, MA, USA
[d] Broad Institute of MIT and Harvard, Cambridge, MA, USA
[e] Department of Bioengineering, Northeastern University, Boston, MA, USA
[f] Department of Mathematics, Northeastern University, Boston, MA, USA
[g] Department of Chemical Engineering, Northeastern University, Boston, MA, USA

ARTICLE INFO

Keywords:
Mechanistic models
Machine learning
Neural networks
Resource competition in synthetic biology

ABSTRACT

This work explores the integration of machine learning (ML) and mechanistic models (MM). While ML has demonstrated remarkable success in data-driven modeling across engineering, biology, and other scientific fields, MM remain essential for their interpretability and capacity to extrapolate beyond observed conditions based on established principles such as chemical kinetics and physiological processes. However, MM can be labor-intensive to construct and often rely on simplifying assumptions that may not fully capture real-world complexity. It is thus desirable to combine MM and ML approaches so as to enable more robust predictions, enhanced system insights, and improved handling of sparse or noisy data. A key challenge when doing so is ensuring that ML components do not disregard mechanistic information, potentially leading to overfitting or reduced interpretability. To address that challenge, this paper introduces the idea of Partially Uncertain Model Structures (PUMS) and investigates conditions that discourage the ML components from ignoring mechanistic constraints. This work also introduces the concept of embedded Physics-Informed Neural Networks (ePINNs), which consist of two loss-sharing neural networks that seamlessly blend ML and MM components. This work arose in the study of the context problem in synthetic biology. Engineered genetic circuits may exhibit unexpected behavior in living cells due to resource sharing. To illustrate the advantages of the ePINNs approach, this paper applies the framework to a gene network model subject to resource competition, demonstrating the effectiveness of this hybrid modeling approach in capturing complex system interactions while maintaining physical consistency.

1. Introduction

1.1. Machine learning or/and mechanistic models?

Machine learning (ML) approaches have advanced rapidly in recent years, demonstrating remarkable success in extracting complex patterns from large datasets and driving innovation across numerous fields [1,2]. In systems and synthetic biology, in particular, ML can be integrated into genetic circuit design, providing new approaches to designing and optimizing synthetic biological constructs [3]. However, purely data-driven ML models often lack explanatory power and can struggle in scenarios where data is limited or do not adequately capture the underlying physics or biology of a process [4]. In contrast, mechanistic models (MM) are grounded on established principles, such as fluid dynamics equations, chemical reaction kinetics, or physiological processes. In systems and synthetic biology, "physics-based" models play a key role when engineering cellular control systems [5]. While they offer interpretability, reliability, and the ability to extrapolate beyond observed conditions [6], such models can be labor-intensive to construct, and necessarily rely on simplifying assumptions that will not always capture the full complexity of real-world systems.

* Corresponding author.
E-mail addresses: atefedarabi93@gmail.com (A. Darabi), z.an199401@gmail.com (Z. An), malirdwi@gmail.com (M.A. Al-Radhawi), willwoncho@gmail.com (W. Cho), m.siami@northeastern.edu (M. Siami), e.sontag@northeastern.edu (E.D. Sontag).
[1] Contributed equally.

https://doi.org/10.1016/j.mbs.2026.109649

Available online 4 March 2026

Combinations of ML and MM hold the promise of leveraging the strengths of both approaches while mitigating their respective weaknesses [7,8]. By combining domain knowledge with data-driven algorithms, it should be possible to develop hybrid models that benefit from the interpretability and theoretical rigor of mechanistic frameworks, as well as the adaptability and predictive power of ML methods. Ideally, this synergy will enable more robust predictions, improved insight into system behavior, and the capacity to handle sparse or noisy datasets more effectively. As such, the emerging field of hybrid modeling has garnered increasing interest, with potential applications from systems biology to engineering, physics, and chemistry. To draw upon the best features of both approaches, various versions of "hybrid" ML/MM combined models have been proposed, at least since the mid-1990s, particularly in chemical process control [9,10]. In systems biology, hybrid models have been used more recently to deal with unknown delays in transcription as well as with unknown portions of signaling pathways [11–13]. Hybrid techniques aim to synergize interpretable models based on physicochemical and biochemical knowledge with data-driven statistical learning. A recent review explores and compares approaches to build ML/MM combined models in systems biology [14].

Typically, these previous works have used ML to fit neural network models to physical constraints, or to generate data for either mechanistic or ML models to train from the other. Here, we are interested in exploring a somewhat different aspect, namely how to incorporate *partially uncertain* MM as a part of larger systems which also contain totally unknown components. We emphasize "partially uncertain" because those mechanistic parts of larger models that are perfectly known may in principle be incorporated *a priori* into a model. In this process, the subsequent ML learning phase can in effect "subtract" that known part of the model and fit the residual. (Implementing that approach is not straightforward, and combining ML with perfectly known systems is of great practical interest in itself.) Let us discuss more precisely the problem to be studied.

1.2. Partially uncertain model structures

For concreteness, in this work we deal with systems defined by ordinary differential equation (ODE) models of the following general form:

$$\dot{x} \;=\; F(x,\pi) + G(x) \tag{1}$$

but similar considerations apply to partial differential equation, delay-differential, discrete time, or even stochastic formulations. Here $x = x(t)$ is a vector function of time, taking values in $\mathbb{R}^n$, and dot indicates time derivative. The first term, a parameterized vector field F, depends on x as well as on a finite set of parameters π. It represents a "mechanistic gray box" part of the system, which is perfectly known *except* for the values of the parameters π. (Examples of such parameters might be binding rates in biochemistry, flow rates between body tissues in pharmacology, masses in mechanics, or resistances in electrical circuits.) This is the "interpretable" part of the system. For example, we may have a two-dimensional differential equation whose coordinates describe the position and velocity of a spring for which mass and stiffness constants are not known (the "known unknowns"). The second term, $G(x)$, denotes a "black box" part of the system, about which nothing is *a priori* known (the "unknown unknowns"). Examples of such might be gene transcription or epigenetic effects in cell biology, immune components in mammalian organisms, or interactions among species in ecology. In the spring example, $G(x)$ might represent unmodeled loads, nonlinear friction, or fatigue effects, thus introducing a "correction" or "compensating" term to the physically known mechanism. We will assume that G can be represented by a function belonging to a typically infinite dimensional "universal" function approximation class, such as deep neural networks or other machine learning architectures. One wishes to use experimental data in order to simultaneously find the (usually physically meaningful and interpretable) parameters π and a function G so that (1) is consistent with this data.

Now, one immediate problem with this admittedly vague formulation is that there is no obvious way to avoid the *overcorrection* phenomenon, in which the function G just chooses to *ignore all mechanistic information*. Indeed, suppose that (1) has been found to match experimental data, for a given parameter vector π. Now pick any other parameter $\tilde{\pi}$. Then the model

$$\dot{x} = F(x,\tilde{\pi}) + \widetilde{G}(x)$$

works equally well, where $\widetilde{G}(x) := G(x) + F(x,\pi) - F(x,\tilde{\pi})$. This is true even if the functions $F(x,\pi)$ and $F(x,\tilde{\pi})$ are distinct, so the issue is not one of identifiability (different parameters giving rise to the same function). In other words, using mechanistic information confers no advantage compared to a straightforward ML approach. Thus, in order for the problem to be well-posed, we need to impose some *structure* to the uncertainty, as we will discuss in Section 2.4. First, however, we will discuss a case study.

1.3. Context in synthetic biology

Our work was motivated by the *context problem* in synthetic biology, meaning that engineered genetic circuits may not perform as expected when inserted into living cells. Synthetic biology seeks to engineer biological systems for applications such as biosensing, programmable probiotics, and regenerative medicine. However, a significant challenge in this field is compositional context, the phenomenon where the properties of a biological circuit module change depending on the presence of other modules. This context-dependence leads to unpredictable behavior, making it difficult to design modular and scalable biological systems. Unlike traditional engineering disciplines, where standardized components interact through well-defined physical principles and abstraction layers, biological modules interact in complex and often unknown ways.

Factors such as retroactivity [15], resource sharing, genetic context effects, growth rate feedback, and off-target interactions can disrupt intended circuit functions [5,15,16] and even result in paradoxical effects in which activators of gene expression in effect function as repressors [17]. Moreover, unlike in other engineering fields where methods like feedback controllers can insulate modules from external influences, synthetic biology lacks robust frameworks to manage these context effects. As a result, genetic circuits are often designed in an ad hoc manner, requiring labor-intensive re-characterization whenever a new component is introduced. This trial-and-error approach slows down progress and limits the scalability of synthetic biology. Understanding context effects is therefore crucial for developing a more systematic and predictable design framework that accounts for unintended interactions and enables the reliable composition of biological modules. Interestingly, such context effects appear to also play a central role in naturally occurring, not synthetic, cellular processes such as for example those involved in epigenetic regulation of metastasis in cancer [18].

We will use a resource-sharing model from synthetic biology as a case study to illustrate our ideas. Specifically, we will work with the four-dimensional model presented in Section 2.2, which is a simplified version of a more detailed model also described there.

1.4. An approach: embedded physics-informed neural networks (ePINNs)

Physics-Informed Neural Networks (PINNs) are deep learning models that integrate physical laws, for example expressed as differential equations, into the training process. Unlike traditional neural networks that rely solely on labeled data, PINNs embed governing equations as constraints in the loss function. This approach has been successfully applied in fields such as fluid dynamics, material science, and biomedical engineering, where data is sparse or expensive to obtain [19]. By leveraging automatic differentiation and modern optimization techniques, PINNs provide a mesh-free alternative to conventional numerical solvers while maintaining physical consistency [20]. PINNs build upon earlier work such as [21], which trained neural networks to approximate PDE solutions by minimizing residuals. PINNs extend this idea by incorporating recent advances in deep learning to enhance accuracy, scalability, and robustness.

Despite their versatility, standard PINNs require the full mechanistic structure of the governing equations to be specified in advance and thus cannot recover genuinely unknown interactions. They are therefore limited in settings where the governing dynamics is partially characterized, which are common in biological systems. In such cases, PINNs may misattribute error by distorting parameter estimates or trajectory reconstructions to compensate for missing dynamics. They also show sensitivity to noise and partial observability, as training typically assumes direct measurements of all relevant states. In addition, they provide no identifiability guarantees, thus offering no formal conditions under which hidden dynamics could in principle be recovered. These limitations restrict their use in contexts where knowledge is fragmentary and data are incomplete, motivating methods that can handle partially uncertain model structures.

To address some of these challenges, *Universal Physics-Informed Neural Networks (UPINNs)* extend PINNs by introducing an auxiliary neural network to represent hidden operators. [22] and [23] demonstrated that UPINNs ability to recover missing terms from sparse, noisy data and express them in symbolic form using AI Feynman regression. [24,25] further advanced the framework by incorporating dimensional analysis to improve efficiency and interpretability, and by applying UPINNs to pharmacology to model chemotherapy drug action. Collectively, these studies highlight the promise of augmenting mechanistic models with learnable components, while questions of identifiability under partial observability remain open. We wish to emphasize that if parameters are known, at least conceptually the problem becomes far easier, since the known part of the system can be "subtracted" and only residuals need to be fit. This motivated our preliminary work in [26,27] and this paper.

In this work, we extend the PINNs idea through the concept of *embedded PINNs (ePINNs)*, originally proposed in [26,27], which is a novel framework composed of two loss-sharing neural networks designed to improve predictions of unknown dynamics from quantitative time-series data. While ePINNs share similarities with UPINNs in jointly learning missing dynamics alongside PINN surrogates, our contribution lies in three main directions: (1) we introduce the PUMS formalism and prove an identifiability theorem that allows one to distinguish the effects from unknown parameters from the effects of unknown dynamics, (2) we use techniques from the theory of observability under unknown inputs in order to understand when both parameters and unknown terms can be recovered under partial observability, and (3) we apply this framework to a synthetic biology gene circuit resource-sharing problem characterized by partial and noisy measurements. We believe that ePINNs should be useful in the analysis of biological systems that are partially understood, addressing unknown interactions, hard-to-measure reaction rates, and missing data. Section 2.1 describes ePINNs, and we later apply ePINNs to the gene circuit resource-sharing problem.

1.5. Outline of paper

The paper is organized as follows. In the methods Section 2 we introduce the biological example to be used as a case study of resource competition in synthetic biology, as well as the ePINNs method and related algorithms and loss function design. We also introduce there a theoretical framework to address the overcorrection problem. The results Section 3 presents simulations, applying ePINNs to infer unknown reaction rates and predict unknown dynamics for the case study, and providing a mathematical result that helps to understand why our example is a good candidate for ePINNs modeling.

The paper ends with a discussion in Section 4.

2. Methods

2.1. Definition of embedded physics-informed neural networks (ePINNs)

In general, we will consider systems as in (1), together with a fixed choice of initial state $x(0) = x_0$ and an output $y = h(x)$. The measurable variables $y = (y_1, \dots, y_l)$ are related to the system states through an output function $h : \mathbb{R}^n \to \mathbb{R}^l$. In typical applications, h is linear. We denote the n-dimensional Euclidean real space by $\mathbb{R}^n$, and the set of $n \times m$ matrices with real elements by $\mathbb{R}^{n\times m}$. For a matrix M, its transpose is denoted as $M^\top$.

As discussed in Section 1.4, this work relies upon embedded PINNs (ePINNs). We introduce ePINNs in this section.

ePINNs combine two neural networks. The first network, which we will denote by $\mathcal{X}$, is parameterized by a vector of weights θ_1. This network attempts to build an estimate of the state trajectory $x(t)$ as a function of the time t and the initial condition x_0. We write the estimated solution $\hat{x}(t)$ as $\hat{x}(t, x_0, \theta_1)$ when we want to be explicit about the dependence on x_0 and θ_1. The second network, which we denote by $\mathcal{G}$, is parameterized by θ_2, and it is intended to model the unknown component $G(x)$ of the dynamics. In summary, the system in (1) will be approximated as follows:

$$\dot{\hat{x}} \;=\; F(\hat{x}, \hat{\pi}) + \mathcal{G}(\hat{x}, \theta_2), \qquad \hat{x}(t) = \hat{x}(t, x_0, \theta_1). \tag{2}$$

We collect the three vector parameters to be estimated into $\Theta = (\pi, \theta_1, \theta_2)$.

The structure of ePINNs is depicted in Fig. 1. It represents a modular architecture consisting of two interconnected neural networks, $\mathcal{X}$ and $\mathcal{G}$. The network $\mathcal{X}$ takes as input the time t and initial state x_0, has a number of layers, and ends in an output layer predicting the state $\hat{x}(t)$. Automatic differentiation (not numerical differentiation, so exact) is employed in order to compute the time derivative $\frac{d\hat{x}}{dt}$, enabling direct evaluation of the governing ODEs. The network $\mathcal{G}$ takes as input the predicted states $\hat{x}(t)$ and, through a feedforward multilayer architecture, produces as its output an approximation $\mathcal{G}(\hat{x}, \theta_2)$ of the unknown dynamics $G(x)$.

2.1.1. Overall loss function structure

The two networks do not interact directly, but they both contribute to the ODE loss function $\mathcal{L}^{\mathrm{ode}}(\theta_1, \theta_2, \hat{\pi})$, which, together with a penalty on the mismatch between $\hat{x}(0)$ and x_0, will be minimized so as to ensure that $\hat{x}(t)$ solves (2). To consistently train both neural networks using the governing ODEs and experimental data, we design a shared loss function for them, with the following components:

1. *Data constraint*, $\mathcal{L}^{\mathrm{d}}(\theta_1)$: Ensures that state predictions from $\mathcal{X}$ match experimental observations.
2. *ODE constraint*, $\mathcal{L}^{\mathrm{ode}}(\theta_1, \theta_2, \hat{\pi})$: Ensures that derivative predictions $\dot{\hat{x}}$, combined with unknown dynamics from $\mathcal{G}$, satisfy the ODE system.
3. *Initial condition constraint*, $\mathcal{L}^{\mathrm{IC}}(\theta_1)$: Ensures that the initial state predictions $\hat{x}(0)$ match the true initial conditions of all states (both observed and unobserved). This loss term is particularly crucial during the training process with partially observed dynamics, as the solution space of the known ODEs would otherwise span the entire $\mathbb{R}^n$ space, unless purposefully guided towards the true initial condition.

During training, we are given data $y_i = y(t_i)$ measured at a finite number of time points $t = (t_1, \dots, t_N)$. For numerical experiments to evaluate the framework, we use a "ground truth" system of the form (1) to generate $y(t)$, and we wish to estimate π (denoted by $\hat{\pi}$) as well as find a network $\mathcal{G}$ that approximates G. We assume that x_0 is known, which is a realistic assumption in many problems, such as our case study below. The data loss term will be checked at the sampling times t_i. We found in practice that it is best to employ an adaptive time sampling frequency which is higher during fast transition periods and is reduced during steady states, enabling effective separation of fast and slow dynamics.

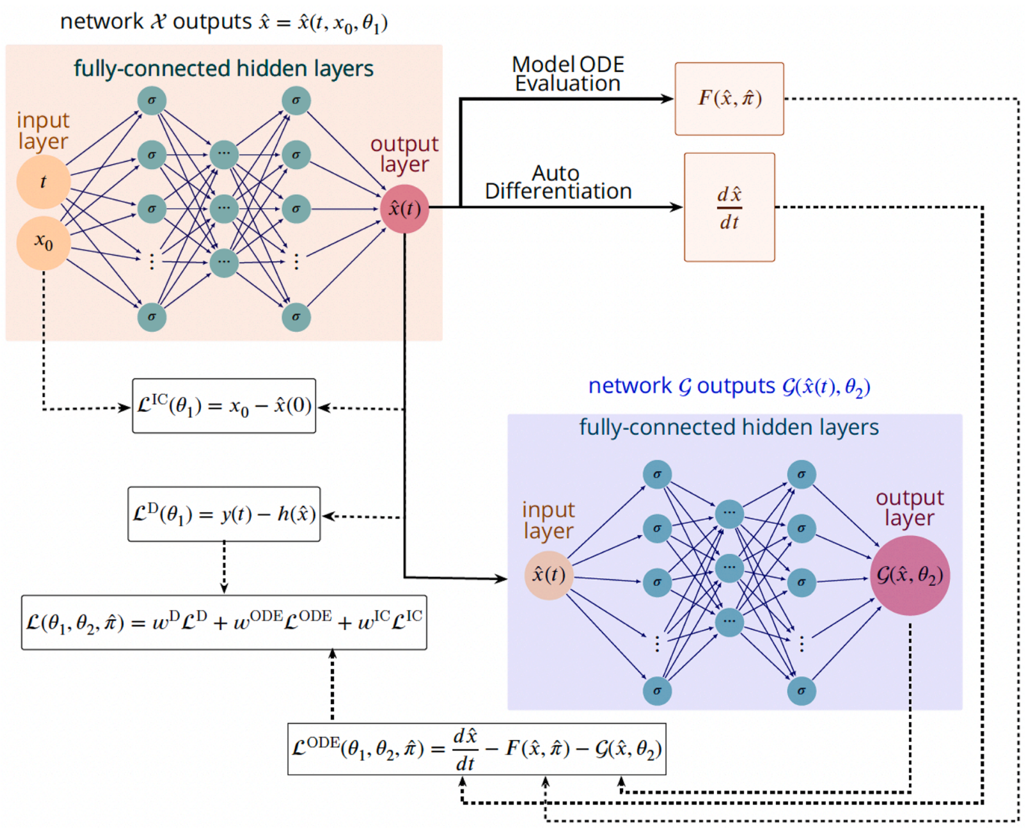
network $\mathcal{X}$ outputs $\hat{x} = \hat{x}(t, x_0, \theta_1)$
fully-connected hidden layers
input layer
t
x_0
output layer
$\hat{x}(t)$
Model ODE Evaluation
$F(\hat{x}, \hat{\pi})$
Auto Differentiation
$\frac{d\hat{x}}{dt}$
$\mathcal{L}^{\mathrm{IC}}(\theta_1) = x_0 - \hat{x}(0)$
$\mathcal{L}^{\mathrm{D}}(\theta_1) = y(t) - h(\hat{x})$
$\mathcal{L}(\theta_1, \theta_2, \hat{\pi}) = w^{\mathrm{D}}\mathcal{L}^{\mathrm{D}} + w^{\mathrm{ODE}}\mathcal{L}^{\mathrm{ODE}} + w^{\mathrm{IC}}\mathcal{L}^{\mathrm{IC}}$
network $\mathcal{G}$ outputs $\mathcal{G}(\hat{x}(t), \theta_2)$
fully-connected hidden layers
input layer
$\hat{x}(t)$
output layer
$\mathcal{G}(\hat{x}, \theta_2)$
$\mathcal{L}^{\mathrm{ODE}}(\theta_1, \theta_2, \hat{\pi}) = \frac{d\hat{x}}{dt} - F(\hat{x}, \hat{\pi}) - \mathcal{G}(\hat{x}, \theta_2)$

Fig. 1. ePINNs architecture. ePINNs structure is composed of two embedded neural networks. The first one, $\mathcal{X}$, has weights θ_1, postulates a trajectory $\hat{x}(t)$ as a function of time and initial conditions $x(0)$. The second one, $\mathcal{G}$, has weights θ_2 and takes as its input the postulated trajectory $\hat{x}(t)$ produced by $\mathcal{X}$, producing a "correction term" $\mathcal{G}(\hat{x}, \theta_2)$ representing totally unknown dynamics. For simplicity, norms and sums over training data are not shown for the various loss functions $\mathcal{L}$; precise definitions are given in the text.

We introduce coefficients, denoted by w^{d}, w^{ode}, and w^{ic}, respectively, to balance the relative contributions of $\mathcal{L}^{\mathrm{d}}$, $\mathcal{L}^{\mathrm{ode}}$, and $\mathcal{L}^{\mathrm{ic}}$ into an overall loss function:

$$\mathcal{L}(\theta_1, \theta_2, \hat{\pi}) \; := \; w^{\mathrm{d}}\mathcal{L}^{\mathrm{d}}(\theta_1) + w^{\mathrm{ode}}\mathcal{L}^{\mathrm{ode}}(\theta_1, \theta_2, \hat{\pi}) + w^{\mathrm{ic}}\mathcal{L}^{\mathrm{ic}}(\theta_1)\,.$$

The explicit forms of losses are defined as follows:

$$\mathcal{L}^{\mathrm{d}}(\theta_1) \; := \; \frac{1}{N}\sum_{j=1}^{N}\left\| y(t_j) - h(\hat{x}(t_j)) \right\|^2,$$

$$\mathcal{L}^{\mathrm{ode}}(\theta_1, \theta_2, \hat{\pi}) \; := \; \frac{1}{N}\sum_{j=1}^{N}\left\| \frac{d\hat{x}}{dt}(t_j) - F(\hat{x}(t_j), \hat{\pi}) - \mathcal{G}(\hat{x}(t_j), \theta_2) \right\|^2,$$

$$\mathcal{L}^{\mathrm{ic}}(\theta_1) \; := \; \left\| x(0) - x_0 \right\|^2.$$

The loss functions are also shown in Fig. 1. The loss components $\mathcal{L}^{\mathrm{d}}(\theta_1)$ and $\mathcal{L}^{\mathrm{ic}}(\theta_1)$ deal with mismatches between model predictions and observations y. On the other hand, the component $\mathcal{L}^{\mathrm{ode}}(\theta_1, \theta_2, \hat{\pi})$ is not directly dependent on data and only relies upon ODE evaluations and automatic differentiation (AD), which computes $\frac{d\hat{x}}{dt}$ [28,29]. AD calculates the gradients of the neural network outputs with respect to their inputs, ensuring accurate and efficient derivative estimation without explicit numerical differentiation.

Training is conducted using gradient-based optimization methods, such as the Adam optimizer [30], to update the parameters θ_1, θ_2, and $\hat{\pi}$ simultaneously. The optimized parameters $(\theta_1^*, \theta_2^*, \pi^*)$ are obtained by minimizing the shared loss function $\mathcal{L}(\theta_1, \theta_2, \hat{\pi})$, as follows:

$$\theta_1^*, \theta_2^*, \pi^* = \underset{\theta_1, \theta_2, \hat{\pi}}{\operatorname{argmin}}\, \mathcal{L}(\theta_1, \theta_2, \hat{\pi}).$$

This simultaneous optimization ensures that the known dynamics, unknown dynamics, and system parameters are cohesively learned during the training process.

2.2. *Case study: a resource competition model in synthetic biology*

As described in the introductory Section 1.3, this work was motivated by the fact that engineered genetic circuits may not perform as expected when inserted into living cells, due to resource sharing.

In systems biology, one considers models in which internal states represent the concentrations of promoters, mRNAs, proteins, and other cellular players. Suppose that we are interested in a synthetic circuit in which a particular gene ultimately leads, through transcription, translation, and post-translational modifications, to the expression of a desired "output" protein, which we will denote by Y. In applications of synthetic biology, Y is a protein to be manufactured by a cell. For example, insulin is such a protein, in fact a small peptide hormone. Recombinant insulin is a synthetic form of human insulin that is used to treat diabetes mellitus by regulating blood sugar levels. Recombinant insulin is made by inserting the human insulin gene into genetically engineered bacteria (*E. coli*) or yeast (*S. cerevisiae*). These microorganisms produce insulin that is harvested and purified for pharmaceutical use.

We view the total concentration of the input gene as an "input" to the gene expression component, which we denote by U and assume is constant. Ideally, after the system settles to a steady state, the output concentration of Y will stabilize to a desired value. The process that leads from U into Y is shown in the green box in Fig. 2.

Fig. 11 presents the predictions for all state variables. Although only Y, M, and M_I were used for training, the model is able to recover the full state trajectories by leveraging the system dynamics.

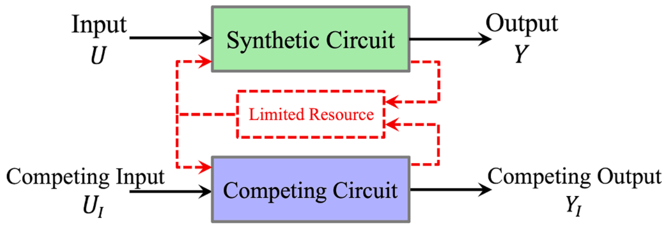

Fig. 2. A synthetic circuit with one competing circuit.

Table 1
State variables for model with two circuits sharing RNAP and ribosomes.

Synthetic Circuit	Competing Circuit	Resources
U: free promoter	U_I: free promoter	R: ribosome
E: promoter-RNAP complex	E_I: promoter-RNAP complex	Q: RNAP
M: mRNA	M_I: mRNA	
F: mRNA-ribosome complex	F_I: mRNA-ribosome complex	
Y: output protein	Y_I: output protein	

Finally, Fig. 12 demonstrates ePINNs' ability to infer the unknown dynamics $G(x)$. The predicted curve aligns well with the ground truth, despite the presence of noise and partial observations.

We prefer to think of U as the promoter associated to the given gene, because the promoter may be free or it might be bound to RNA polymerase (RNAP) molecules. We will denote the concentration of the unbound (free) promoter by U. RNAPs are the "machines" that transcribe into an intermediate component, mRNA, the genetic information encoded in DNA. Subsequently, mRNA molecules guide the production of the protein Y, a process that is facilitated by ribosomes, which are the "machines" that help translate mRNA information into aminoacid sequences. Such a gene transcription process typically shares the use of one or more limited resources (shown as a red box in the figure) which are also used by other "competing" (or "interfering") genes (denoted by subscripts I). These cellular resources include notably ribosomes (R in the model to be described) and RNA polymerase (RNAP) (Q). For simplicity, let us lump all such competing circuits into one (blue box in the figure). We will subscript I to denote components of this other circuit; it has input U_I and produces protein Y_I.

As a convention, we will use italicized variables (such as M) to denote concentrations in differential equations, and use roman variables (such as M) to denote the corresponding chemical species. With this convention, a simple model of the chemical reactions associated with the two-gene competing system is defined as follows:

$$\mathrm{U}+\mathrm{Q}\underset{k_{12}}{\overset{k_{11}}{\rightleftharpoons}}\mathrm{E}\xrightarrow{\alpha_{11}}\mathrm{U}+\mathrm{Q}+\mathrm{M},\ \mathrm{M}\xrightarrow{\alpha_{12}}\varnothing$$
$$\mathrm{M}+\mathrm{R}\underset{\omega_{12}}{\overset{\omega_{11}}{\rightleftharpoons}}\mathrm{F}\xrightarrow{\beta_{1}}\mathrm{M}+\mathrm{R}+\mathrm{Y},\ \mathrm{Y}\xrightarrow{\beta_{2}}\varnothing$$
$$\mathrm{U_I}+\mathrm{Q}\underset{k_{22}}{\overset{k_{21}}{\rightleftharpoons}}\mathrm{E_I}\xrightarrow{\alpha_{21}}\mathrm{U_I}+\mathrm{Q}+\mathrm{M_I},\ \mathrm{M_I}\xrightarrow{\alpha_{22}}\varnothing$$
$$\mathrm{M_I}+\mathrm{R}\underset{\omega_{22}}{\overset{\omega_{21}}{\rightleftharpoons}}\mathrm{F_I}\xrightarrow{\beta_{1}}\mathrm{M_I}+\mathrm{R}+\mathrm{Y_I},\ \mathrm{Y_I}\xrightarrow{\beta_{2}}\varnothing.$$

The first two sets of reactions describe the transcriptional and translational processes of the synthetic circuit, including degradation/dilution processes, while the last two sets describe the same processes for the competing circuit. There are five state variables for each circuit: U, E, M, F, Y for the synthetic circuit, and U_I, E_I, M_I, F_I, Y_I for the competing circuit. Additionally, there are two shared cellular resources, R and Q, resulting in a 12-dimensional full system. The state variables corresponding to each species are summarized in Table 1.

A mass action kinetics representation of this system, including kinetic constants, is as follows.

$$\begin{aligned}
\dot{Q} &= -k_{11}UQ+(k_{12}+\alpha_{11})E-k_{21}U_IQ+(k_{22}+\alpha_{12})E_I\\
\dot{R} &= -w_{11}MR+(w_{12}+\beta_1)F-w_{21}M_IR+(w_{22}+\beta_1)F_I\\
\dot{U} &= -k_{11}UQ+(k_{12}+\alpha_{11})E\\
\dot{E} &= k_{11}UQ-(k_{12}+\alpha_{11})E\\
\dot{M} &= \alpha_{11}E-\alpha_{12}M-w_{11}MR+(w_{12}+\beta_1)F\\
\dot{F} &= w_{11}MR-(w_{12}+\beta_1)F\\
\dot{Y} &= \beta_1F-\beta_2Y\\
\dot{U}_I &= -k_{21}U_IQ+(k_{22}+\alpha_{21})E_I\\
\dot{E}_I &= k_{21}U_IQ-(k_{22}+\alpha_{21})E_I\\
\dot{M}_I &= \alpha_{21}E_I-\alpha_{22}M_I-w_{21}M_IR+(w_{22}+\beta_1)F_I\\
\dot{F}_I &= w_{21}M_IR-(w_{22}+\beta_1)F_I\\
\dot{Y}_I &= \beta_1F_I-\beta_2Y_I\,.
\end{aligned}$$

It is easy to see from the equations that the following concentrations remain constant over time:

$$\begin{aligned}
U+E &\equiv U_T\\
U_I+E_I &\equiv U_{I,T}\\
R+F+F_I &\equiv R_T\\
Q+E+E_I &\equiv Q_T
\end{aligned}$$

(for example, the time derivative of $U(t)+E(t)$ is zero) so these represent conservation laws that allow one to reduce the equations (given a particular set of initial conditions) to an 8-dimensional system. The subscripts "T" stand for "total." Natural initial conditions are $Q(0)=Q_T$, $R(0)=R_T$, $U(0)=U_T$, $U_I(0)=U_{I,T}$, with the remaining variables set to zero. Using these conservation laws, we may eliminate variables as follows:

$$\begin{aligned}
E &:= U_T-U\\
E_I &:= U_{I,T}-U_I\\
F_I &:= R_T-R-F\\
Q &:= Q_T-E-E_I
\end{aligned}$$

and we obtain the following set of equations, which we will call the "8D" model:

$$\begin{aligned}
\dot{R} &= -w_{11}MR+(w_{12}+\beta_1)F-w_{21}M_IR+(w_{22}+\beta_1)(R_T-R-F)\\
\dot{U} &= -k_{11}U(Q_T-E-E_I)+(k_{12}+\alpha_{11})(U_T-U)\\
\dot{M} &= \alpha_{11}(U_T-U)-\alpha_{12}M-w_{11}MR+(w_{12}+\beta_1)F\\
\dot{F} &= w_{11}MR-(w_{12}+\beta_1)F\\
\dot{Y} &= \beta_1F-\beta_2Y\\
\dot{U}_I &= -k_{21}U_I(Q_T-E-E_I)+(k_{22}+\alpha_{21})(U_{I,T}-U_I)\\
\dot{M}_I &= \alpha_{21}(U_{I,T}-U_I)-\alpha_{22}M_I-w_{21}M_IR+(w_{22}+\beta_1)F_I\\
\dot{Y}_I &= \beta_1(R_T-R-F)-\beta_2Y_I\,.
\end{aligned}$$

RNAP and ribosome competition give rise to different behaviors, studied for example in [31]. Let us focus on the most common scenario of limited ribosome availability but abundant RNAP. In this case, we assume that R is the bottleneck resource, while Q is maintained at a constant value $Q(t)\equiv Q_T$. As a simplified model in that case, in the 12D system we use only the first three conservation laws, and we drop the differential equation for $Q(t)$, replacing Q by Q_T in all the equations. This results in a "abundant-Q_T 8D" model (not shown). Next note that Y_I does not affect the first 7 equations. Moreover, in our application, we will assume that Y_I cannot be measured, as it represents protein produced by unknown genes. The model obtained by dropping the Y_I equation from the abundant-Q_T 8D model will be called the "7D" system:

$$\begin{aligned}
\dot{R} &= -w_{11}MR+(w_{12}+\beta_1)F-w_{21}M_IR+(w_{22}+\beta_1)(R_T-R-F)\\
\dot{U} &= -k_{11}UQ_T+(k_{12}+\alpha_{11})(U_T-U)\\
\dot{M} &= \alpha_{11}(U_T-U)-\alpha_{12}M-w_{11}MR+(w_{12}+\beta_1)F\\
\dot{F} &= w_{11}MR-(w_{12}+\beta_1)F\\
\dot{Y} &= \beta_1F-\beta_2Y
\end{aligned}$$

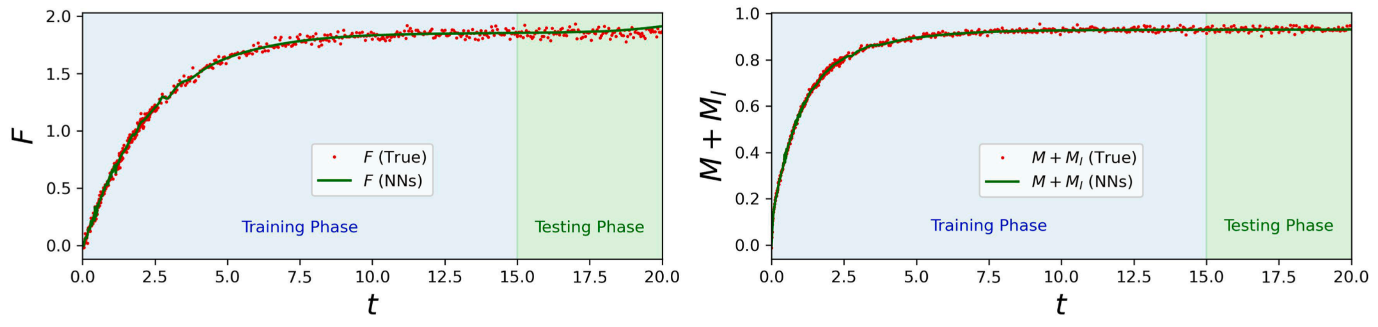

Fig. 3. 4D Model: Physically uninformed neural network ($w^{\text{ODE}} = 0$) **results for observables.** Predictions for state variables R, $M + M_I$ (solid green lines, labeled "NNs") are shown using noisy data from F and $M + M_I$ (red dots, labeled "True"). The physically uninformed network provides an accurate estimation for observed states.

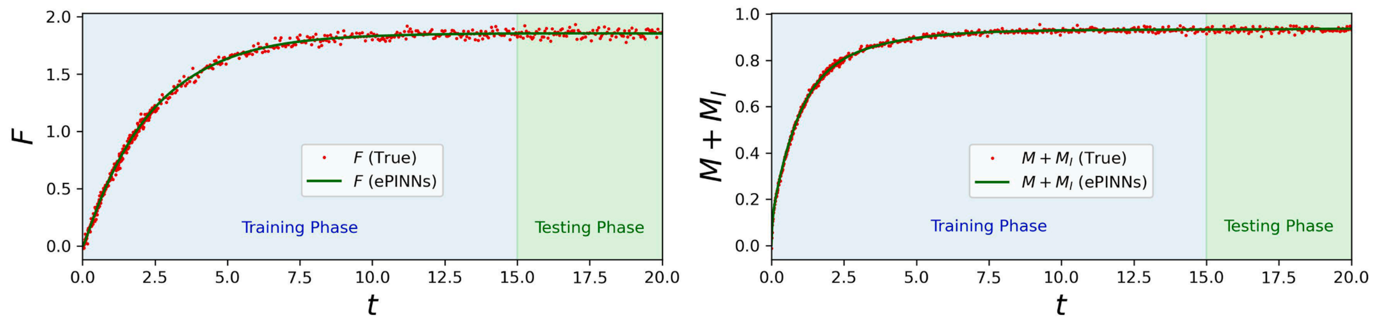

Fig. 4. 4D Model: Physically informed neural network ($w^{\text{ODE}} = 0.5$) **results for observables.** Predictions for state variables F and $M + M_I$ (solid green lines, labeled "ePINNs") are shown using noisy data from F and $M + M_I$ (red dots, labeled "True"). The physically informed network provides an accurate estimation for observed states.

$$\dot{U}_I = -k_{21} U_I Q_T + (k_{22} + \alpha_{21})(U_{I,T} - U_I)$$
$$\dot{M}_I = \alpha_{21}(U_{I,T} - U_I) - \alpha_{22} M_I - w_{21} M_I R + (w_{22} + \beta_1)(R_T - R - F),$$

with initial condition $(R_0, U_0, M_0, F_0, Y_0, U_{I0}, M_{I0})^\top = (R_T, U_T, 0, 0, U_{I,T}, 0)^\top$. We consider the kinetic rates $\pi = (\alpha_{21}, \alpha_{22}, k_{21})$ (marked in blue for emphasis) as unknown model parameters, while the remaining parameters are known. There is also a "black box" competition effect (grayed-out) that impacts the amount of free ribosomes (R) available to M. This is the G part of the system that will be approximated by the neural network $\mathcal{G}$. The vector field "$F(x, \pi)$" corresponds to the non-grayed-out part of the equations. (Not to be confused with F also denoting one of the components of the state $x = (R, U, M, F, Y, U_I, M_I)^\top$.) We will take the output as (Y, M, M_I). Using Y instead of F is more realistic (as this will typically represent a reporter protein, for example) than using the complex F as an output. We assume that we can individually measure M and M_I. Such a measurement could be obtained by labeling the target mRNA M and separately measuring the total amount of transcripts in the cell, so that M_I would be the difference between these two numbers. The structural identifiability analysis of the unknown model parameters under this set of outputs is presented in 2.3.1.

Of course, if we knew that the interaction between M_I and R has this particular form in the gray box, the problem would be reduced to the parameter-fitting problem of finding w_{21}, w_{22}, and β_1 in addition to α_{21}, α_{22}, and k_{21}. However, we want to allow for the possibility of more general and unknown interactions. These might include ribosome stalling, use of rare codons, proofreading steps, ribosomal frameshifting, ribosome pausing and translational regulation, and other processes. Generally, a multi-step translational model may be more appropriate, such as the ribosome flow model [32] and in particular the splitting of the ribosome pool by use of orthogonal ribosomes [33].

A final possible simplification is as follows. Under an assumption of rapid binding and unbinding of RNAP to promoters, we assume an equilibrium value for the free promoters U and U_I, and absorb these into the rates of transcription α_{11} and α_{21} respectively. Thus we arrive to the following "5D" system:

$$\dot{R} = -w_{11} M R + (w_{12} + \beta_1) F - w_{21} M_I R + (w_{22} + \beta_1)(R_T - R - F)$$
$$\dot{M} = \alpha_{11} - \alpha_{12} M - w_{11} M R + (w_{12} + \beta_1) F$$
$$\dot{F} = w_{11} M R - (w_{12} + \beta_1) F$$
$$\dot{Y} = \beta_1 F - \beta_2 Y$$
$$\dot{M}_I = \alpha_{21} - \alpha_{22} M_I - w_{21} M_I R + (w_{22} + \beta_1) F_I$$

The protein Y is typically measured in experiments, so we did not drop the equation for Y in the 7D system. However, in order to study a minimal model, we will drop it from the equations in the 5D model, resulting in a "4D" model to be studied:

$$\dot{R} = -w_{11} M R + w_{12} F - w_{21} M_I R + w_{22}(R_T - R - F)$$
$$\dot{M} = \alpha_{11} - \alpha_{12} M - w_{11} M R + w_{12} F$$
$$\dot{F} = w_{11} M R - w_{12} F$$
$$\dot{M}_I = \alpha_{21} - \alpha_{22} M_I - w_{21} M_I R + w_{22}(R_T - R - F).$$

(We have absorbed β_1 into w_{12} and w_{12}.) The variables are R (free ribosomes), M and M_I (target and competing mRNA), and F (ribosomes bound to M, thought of as production rate of Y). The initial condition is $(R_0, M_0, F_0, M_{I0})^\top = (R_T, 0, 0, 0)^\top$. We view this system as consisting of a partially unknown competing subsystem represented by the M_I variable, in which the unknown parameters are the two kinetic rates $\pi = (\alpha_{21}, \alpha_{22})$ (marked in blue for emphasis). We will use this 4D system in order to numerically explore how partial mechanistic knowledge can improve data fits using ePINNs.

One last issue in setting up the model for numerical experimentation is the choice of output y. Observe from the equations that even if all three of R, M, F were measured, it would be impossible to get any information whatsoever about M_I, since M_I only affects R (and indirectly M and F) only through the totally unknown black box $G(x)$. This means that there is no hope of identifying the unknown parameters. Therefore, to

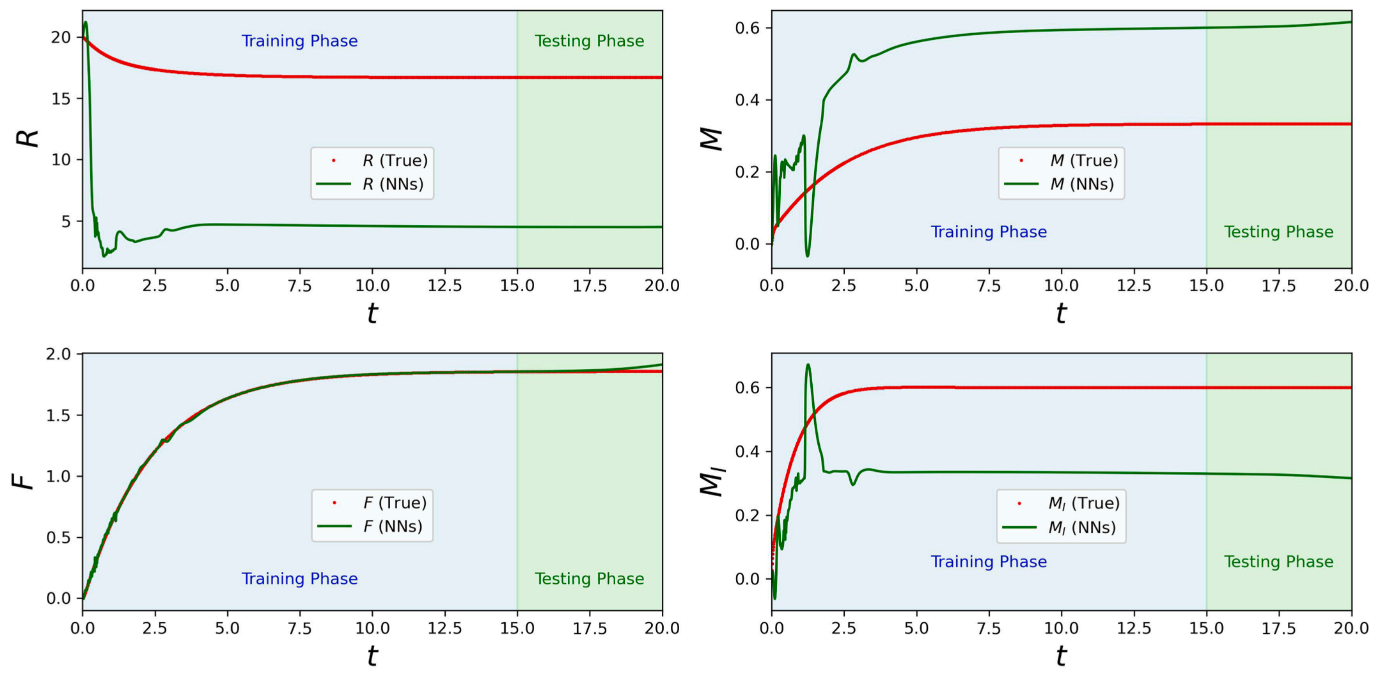

Fig. 5. 4D Model: Physically uninformed neural network ($w^{\text{ODE}} = 0$) **predictions for all states.** Predictions for all state variables (solid green lines, labeled "NNs") are shown using noisy data from F and $M + M_I$ (their true non-noisy values shown in red dots, labeled "True"). The physically uninformed network fails to accurately estimate unobserved states due to the absence of physics knowledge.

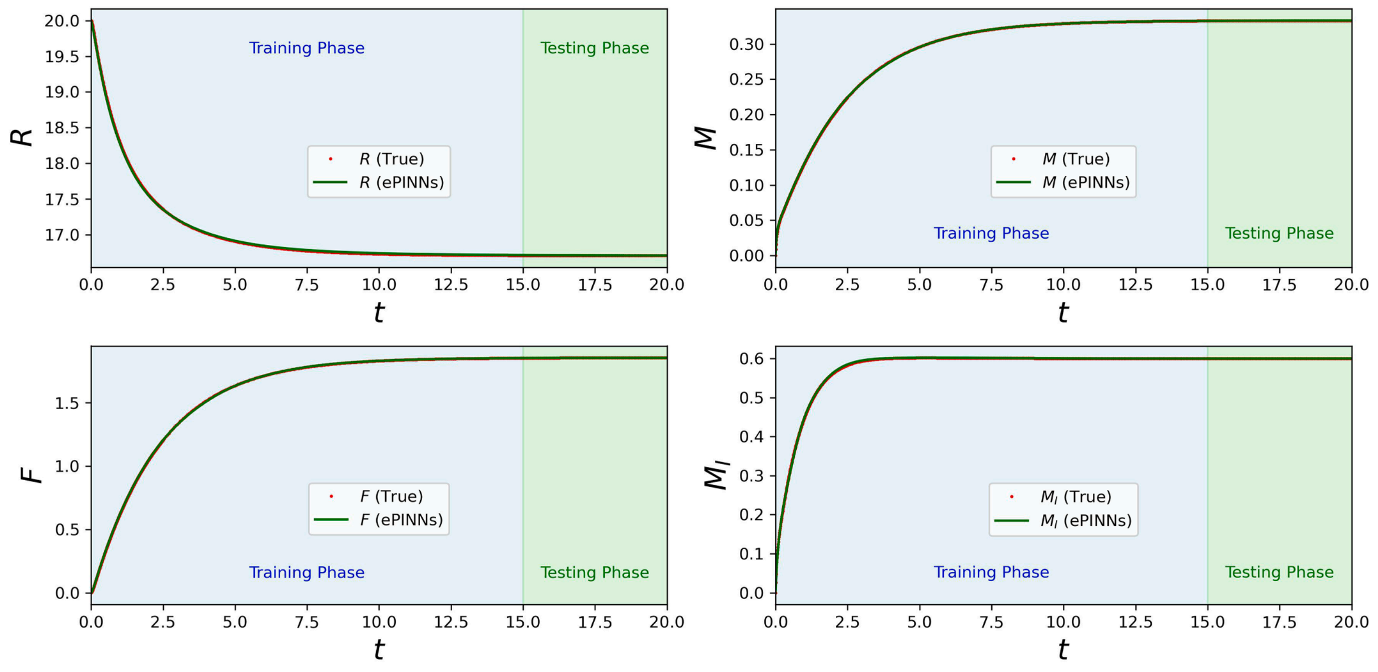

Fig. 6. 4D Model: Physically informed neural network ($w^{\text{ODE}} = 0.5$) **predictions for all states.** Predictions for all state variables (solid green lines, labeled "ePINNs") are shown using noisy data from F and $M + M_I$ (their true non-noisy values shown in red dots, labeled "True"). The physically informed network accurately estimates all states, leveraging system dynamics even in the absence of direct data.

make the problem nontrivial, we have picked the output $y = (F, M + M_I)^\top$. In other words, we assume that we can measure the expressed protein of interest as well as the total amount of mRNAs. The structural identifiability analysis of the unknown model parameters under this set of outputs is presented in 2.3.2.

By focusing on the reduced 7D and 4D systems, we aim to highlight the key concepts more clearly while avoiding unnecessary complexity.

2.3. Structural observability and parameter identifiability analysis

To confirm the feasibility of the hybrid state reconstruction, parameter identification, and hidden mechanism estimation problem-that is, whether ePINNs can successfully recover all states, unknown parameters, and hidden dynamics under the specified output configurations and unknowns for both the 7D and 4D models-we perform a structural

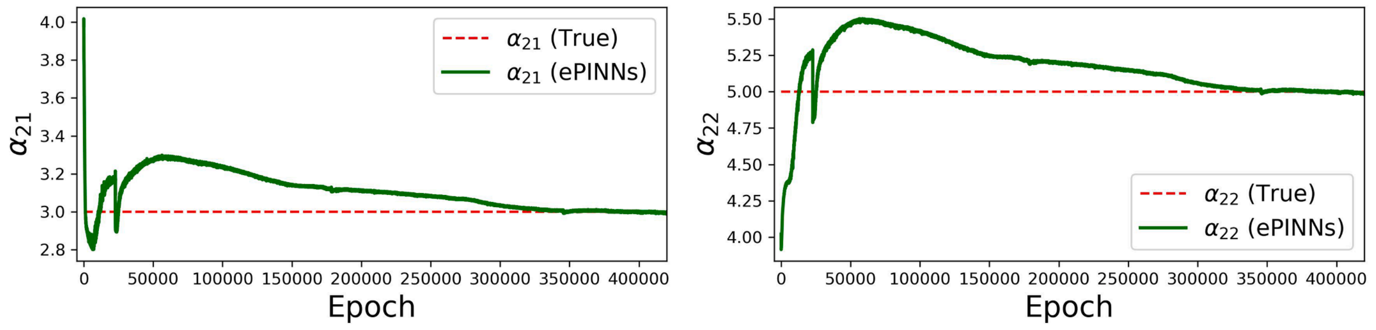

Fig. 7. 4D Model: Learning patterns for unknown parameters α_{21} and α_{22} with partial data and missing dynamics. The solid green lines (labeled “ePINNs”) represent the estimated values, while the dashed red lines (labeled “True”) indicate the true values. The training process uses noisy observations from F and $M + M_I$, demonstrating ePINNs’ ability to identify unknown parameters.

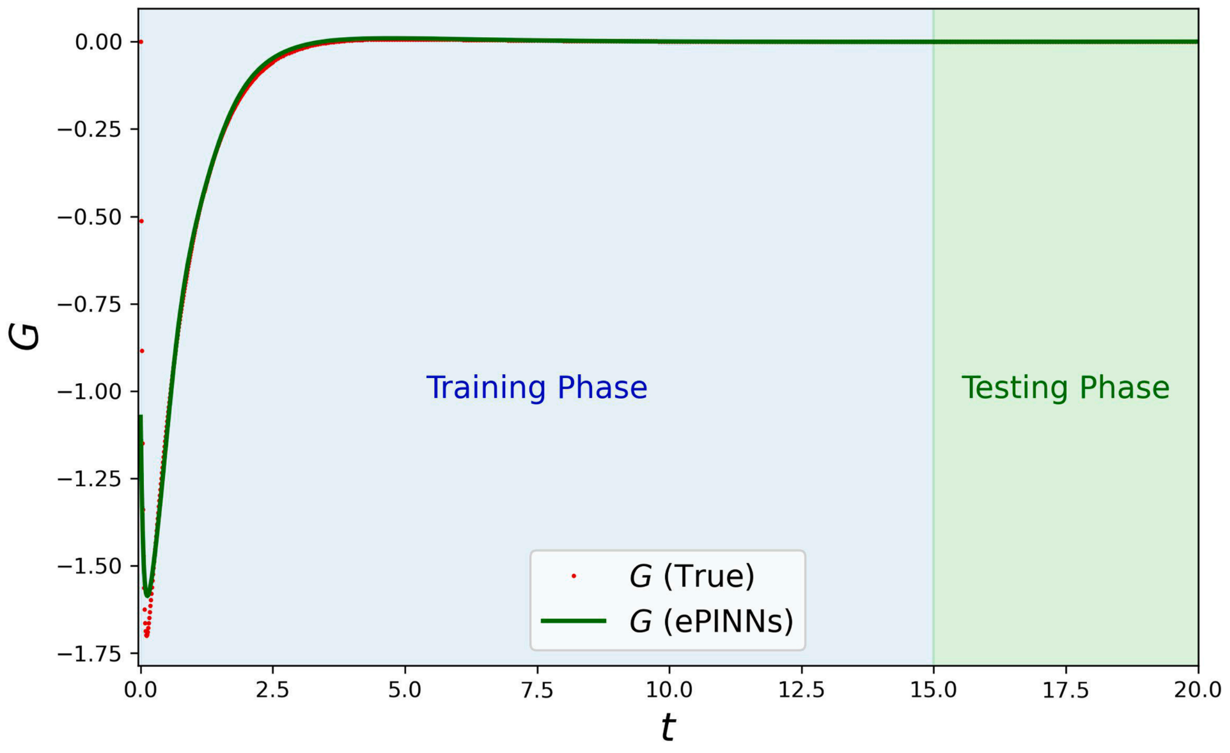

Fig. 8. 4D Model: ePINNs prediction of missing dynamics $G(x)$ with partial data. The solid green line (labeled “ePINNs”) represents the model’s prediction for the unknown dynamics, while red dots (labeled “True”) indicate ground truth values. The training uses noisy observations from F and $M + M_I$, demonstrating the network’s ability to infer hidden system components.

observability and identifiability analysis. The first step consists of thinking of the unknown (neural network) dynamics as "unknown inputs" affecting the system. For example, in the 4D model, this would look like:

$$\begin{aligned}\dot{R} &= -w_{11} M R + w_{12} F + u(t)\\ \dot{M} &= \alpha_{11} - \alpha_{12} M - w_{11} M R + w_{12} F\\ \dot{F} &= w_{11} M R - w_{12} F\\ \dot{M}_I &= \alpha_{21} - \alpha_{22} M_I + u(t).\end{aligned}$$

The next step is to think of $u(t)$ as a polynomial whose coefficients are parameters which are then appended to the unknown parameters of the model. The degree of this polynomial is a hyperparameter, and if identifiability fails, a larger degree is tried. An equivalent way of formulating this approach, which is more suitable for checking using the Observability Rank Condition (ORC) [34], is to add states $u_0, u_1, \ldots,$ and equations $du_i/dt = u_{i+1}$, with the last derivative identically zero. This method was suggested in [35–37]. The ORC extended to unmeasured inputs in this manner, is known as the "Extended Observability Rank Condition" for systems with unmeasured inputs. In this approach, the system is augmented by incorporating the unknown parameters and unmeasured inputs (along with their time derivatives up to a finite order) into an extended state vector. Then, Lie derivatives of the output functions are recursively computed with respect to this augmented system, and the resulting observability-identifiability matrix is constructed from the Jacobians of these Lie derivatives.

We use the STRIKE-GOLDD toolbox [38] to carry out this identifiability test. To address the computational complexity of symbolic differentiation, STRIKE-GOLDD uses a power series-based semi-numerical implementation that relies on Newton iteration and modular arithmetic. This enables efficient rank computation of the observability-identifiability matrix, even for systems with rational nonlinearities and unmeasured inputs, by evaluating the matrix over a finite field. A system is deemed structurally observable if this matrix attains full rank.

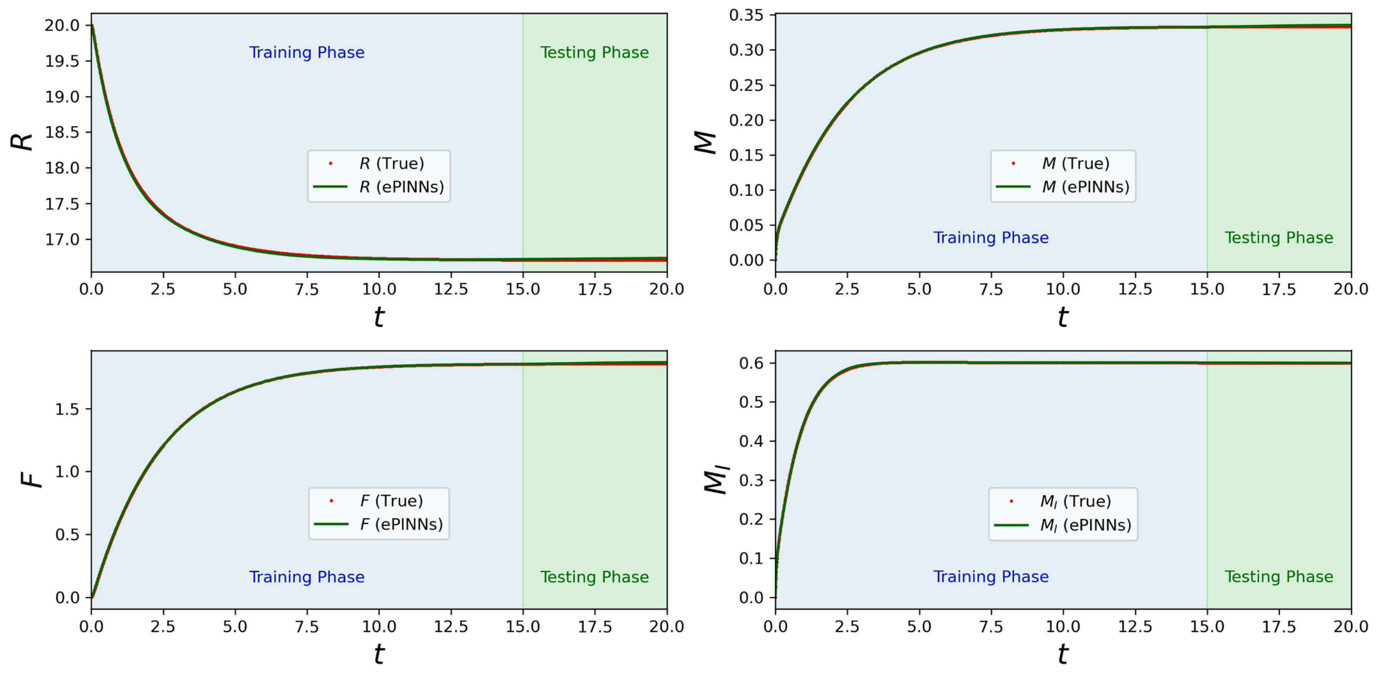

Fig. 9. 4D Model: Predictions for all state variables with partial data, unknown parameters, and missing dynamics. Predictions for all state variables (solid green lines, labeled "ePINNs") are shown using noisy data from F and $M + M_I$ (their true non-noisy values shown in red dots, labeled "True"). The results illustrate the effectiveness of ePINNs in reconstructing unobserved states using physics-informed constraints.

Furthermore, we perform analytical parameter identifiability analysis for each model under its respective output configuration, assuming the unknown dynamics are reconstructible (as verified by STRIKE-GOLDD). This serves to confirm that the chosen set of unknown parameters is structurally identifiable and therefore feasible to infer.

2.3.1. 7D model analysis

We begin with the 7D model, for which STRIKE-GOLDD confirms that all state variables are structurally observable, and the system is structurally identifiable under the proposed output configuration, that is Y, M, and M_I are being observed, even in the presence of a time-varying unmodeled dynamic term. This result justifies the assumption that the unknown dynamics are reconstructible –and hence can be "subtracted" from the model– and thus motivates an identifiability analysis focused on the remaining parametric structure.

For the purpose of this analysis, we consider a broader scenario in which the parameter k_{22} is also treated as unknown. This allows us to gain insight into how the choice of problem formulation affects parameter identification when using ePINNs. To investigate the structural identifiability of the model parameters k_{21}, k_{22}, α_{21}, and α_{22}, we examine a simplified subsystem (presented below) in which only the trajectory of $M_I(t)$ is observed, while $U_I(t)$ remains unmeasured. This choice of subsystem is due to the fact that the unknown parameters appear only in the dynamics of M_I and U_I. For this analysis, we exclude any black-box (unknown) dynamics, focusing solely on the parametric component of the model. Note that, after removing the unknown dynamics, these variables are decoupled from the remaining variables. The governing dynamics will then be:

$$\begin{aligned}\dot{U}_I &= -k_{21}U_I Q_T + (k_{22}+\alpha_{21})(U_{I,T}-U_I),\\ \dot{M}_I &= \alpha_{21}(U_{I,T}-U_I) - \alpha_{22}M_I.\end{aligned}$$

We define the state vector and output as

$$x(t) = \begin{pmatrix} U_I(t) \\ M_I(t) \end{pmatrix}, \quad y(t) = M_I(t).$$

This system can be written in state-space form:

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t),$$

where $u(t) \equiv 1$ and

$$A = \begin{pmatrix} -(k_{21}Q_T + k_{22} + \alpha_{21}) & 0 \\ -\alpha_{21} & -\alpha_{22} \end{pmatrix}, \quad b = \begin{pmatrix} (k_{22}+\alpha_{21})U_{I,T} \\ \alpha_{21}U_{I,T} \end{pmatrix},$$
$$C = \begin{pmatrix} 0 & 1 \end{pmatrix}.$$

The first and second derivatives of the output $y(t)$ are given by:

$$\dot{y}(t) = C\dot{x}(t) = CAx(t) + CBu(t) = -\alpha_{21}U_I - \alpha_{22}M_I + \alpha_{21}U_{I,T},$$

$$\ddot{y}(t) = C\ddot{x}(t) = CA\dot{x}(t) = CA(Ax+Bu) = CA^2x + CABu.$$

Since A and B are constant, $\ddot{y}(t)$ is a linear combination of the state variables U_I and M_I with coefficients involving the model parameters. Substituting back, we can express $\dot{y}(t)$ and $\ddot{y}(t)$ in terms of $y(t)$ and its derivatives:

$$\ddot{y}(t) = -(k_{21}+k_{22}+\alpha_{21})\dot{y}(t) - (k_{21}+k_{22}+\alpha_{21})\alpha_{22}y(t) + \alpha_{21}k_{21}.$$

This implies that only specific combinations-$\alpha_{21}k_{21}$, $k_{21}+k_{22}+\alpha_{21}$, and α_{22}-appear in the output equation. Hence, α_{22} is structurally identifiable, while the individual parameters k_{21}, k_{22}, and α_{21} are not separately identifiable from $M_I(t)$ alone.

It is easy to show that if any one of the parameters k_{21}, k_{22}, or α_{21} is known, the remaining parameters become structurally identifiable. This result is fully consistent with the structural identifiability analysis performed using STRIKE-GOLDD, which confirms that the indeterminacy lies in the combinations of these three parameters when $M_I(t)$ is the only observed output. Therefore, to keep the identifiability problem feasible for ePINNs, we assume that k_{22} is known.

2.3.2. 4D model analysis

Here, we investigate further the observability and identifiability of the 4D model described earlier. Using STRIKE-GOLDD, we model the unknown competition mechanism as a time-varying unmeasured input and examine the system defined by the state variables R, M, F, and M_I, with unknown parameters α_{21} and α_{22}. The outputs are limited to the protein production rate F and the total mRNA concentration $M + M_I$. STRIKE-GOLDD confirms that the observability-identifiability matrix attains full

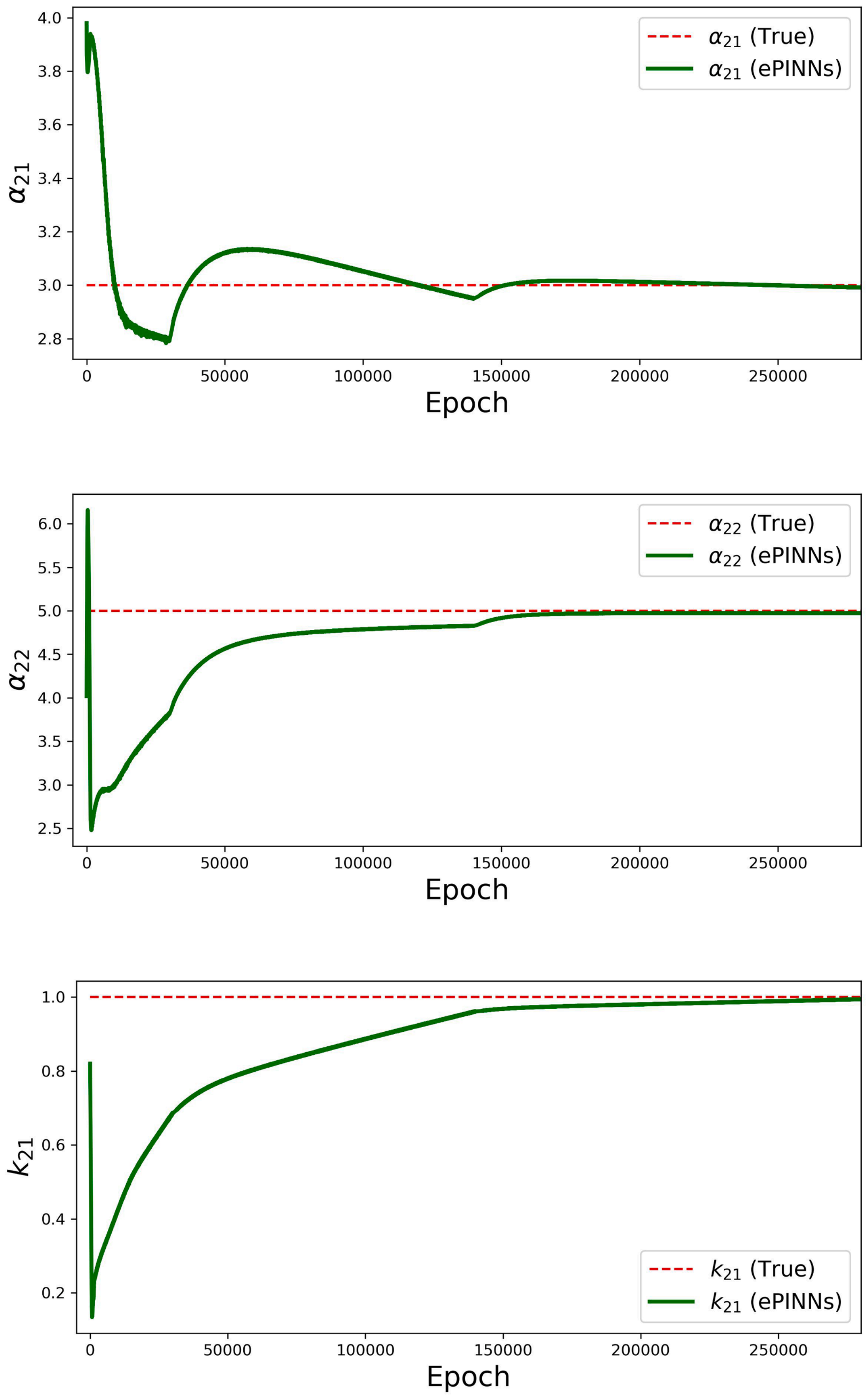

Fig. 10. 7D Model: Learning patterns of unknown parameters α_{21}, α_{22}, and k_{21} using ePINNs. The solid green lines (labeled "ePINNs") represent the estimated values, while the dashed red lines (labeled "True") indicate the true values. The training process uses noisy observations from Y, M, and M_I, demonstrating ePINNs' ability to identify unknown parameters.

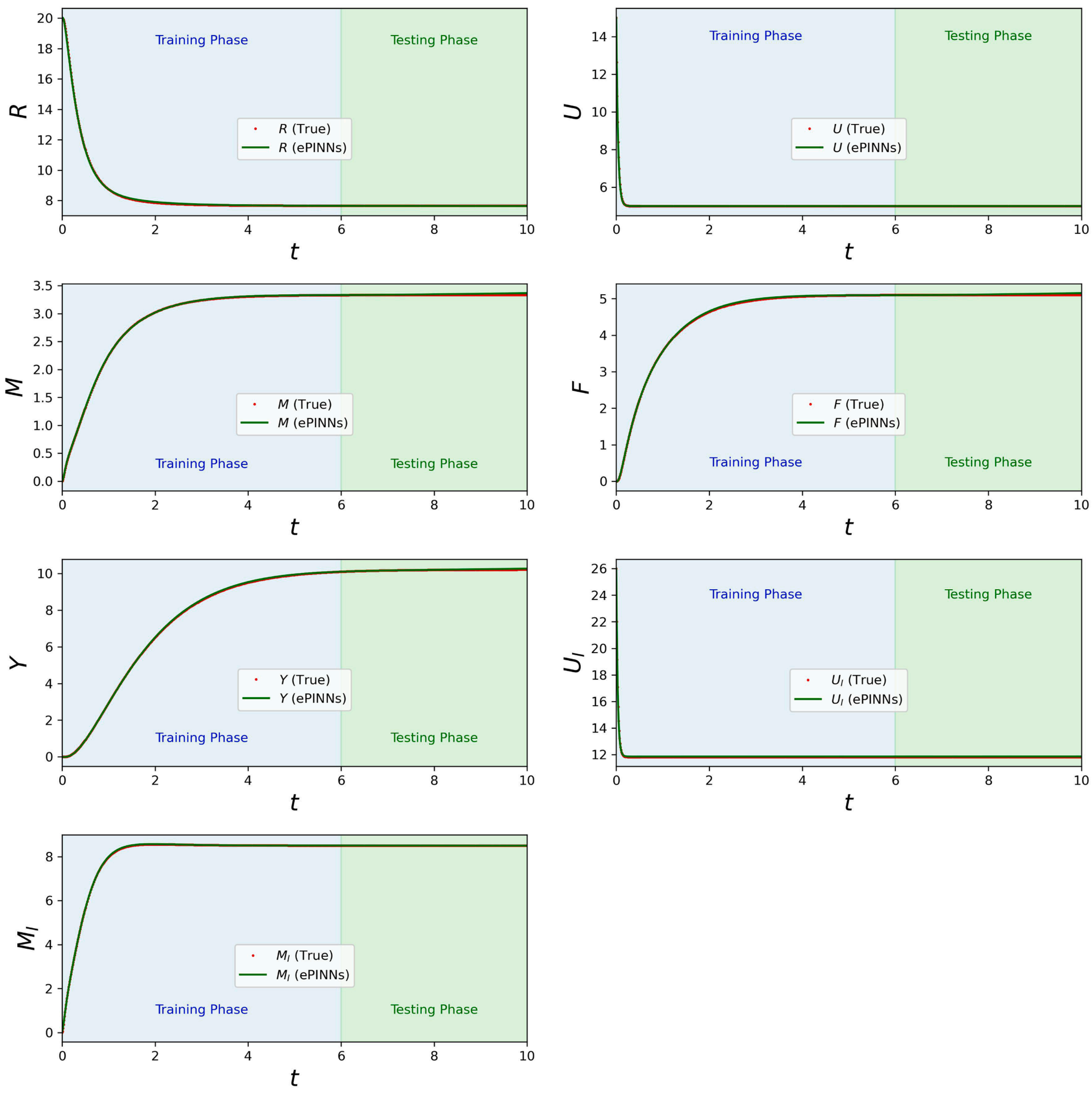

Fig. 11. 7D Model: Predictions for all state variables of the 7D model with partial data and missing dynamics. Predictions for all state variables (solid green lines, labeled "ePINNs") are shown using noisy data from Y, M, and $M + M_I$ (their true non-noisy values shown in red dots, labeled "True"). The results illustrate the effectiveness of ePINNs in reconstructing unobserved states using physics-informed constraints.

rank, indicating that all four state variables and both unknown parameters are structurally observable and identifiable-even in the presence of an unmeasured input.

Based on this result, we assume that the unknown dynamics are reconstructible, and proceed with analytical parameter identifiability analysis. We assume the observed outputs are:

$$y_1 = F, \quad y_2 = M + M_I.$$

The first and second derivatives of the outputs will then be:

$$\dot{y}_1 = w_{11} M R - w_{12} y_1,$$

$$\dot{y}_2 = -\alpha_{12} M - w_{11} M R + w_{12} y_1 - \alpha_{22} M_I + \alpha_{11} + \alpha_{21},$$

and the second derivative is given by:

$$\ddot{y}_2 = -\dot{y}_1 + \alpha_{12} y_2 + \frac{(\alpha_{12} - \alpha_{22})(\dot{y}_2 - \alpha_{11} - \alpha_{21})}{\alpha_{22}}.$$

From this derivative, we observe that α_{21} and α_{22} appear in a rational, coupled form involving only observable outputs (y_1, y_2) and known parameters $(\alpha_{11}, \alpha_{12})$. However, this expression alone does not directly isolate the parameters. To further disentangle the coupling, we compute the third derivative of y_2:

$$\dddot{y}_2 = -\ddot{y}_1 + \alpha_{12} \dot{y}_2 + \frac{(\alpha_{12} - \alpha_{22})}{\alpha_{22}} \ddot{y}_2.$$

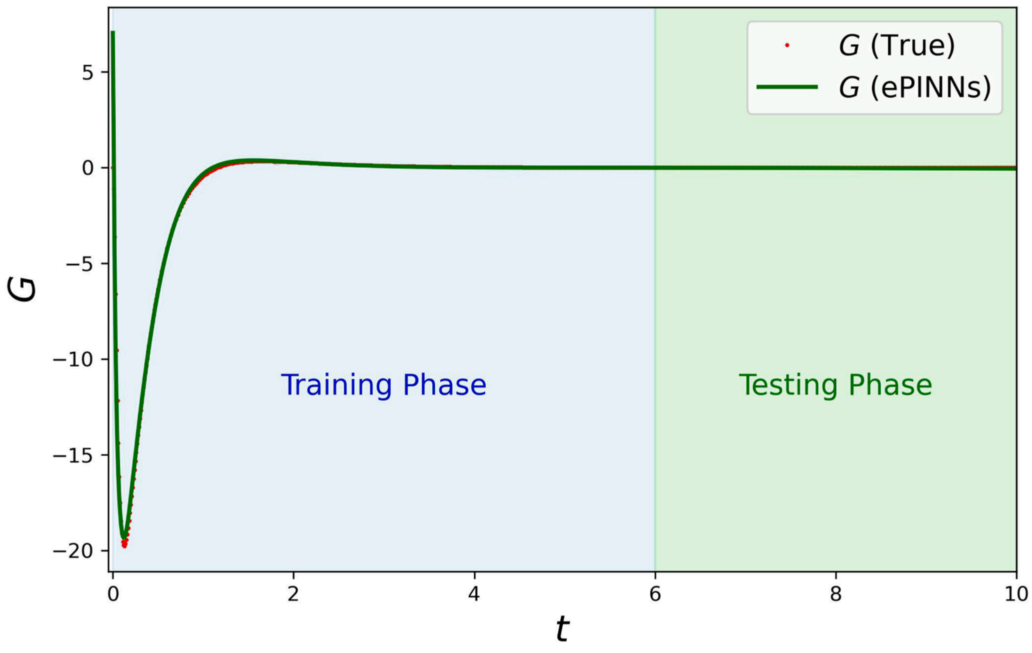

Fig. 12. 7D Model: ePINNs prediction of missing dynamics $G(x)$ with partial data. The solid green line (labeled "ePINNs") represents the model's prediction for the unknown dynamics, while red dots (labeled "True") indicate ground truth values. The training uses noisy observations from Y, M, and M_I, demonstrating the network's ability to infer hidden system components.

This expression depends only on α_{22} (and not on α_{21}), while all other terms are either known or directly measurable. Hence, α_{22} can be uniquely recovered from this equation. Once α_{22} is determined, we can substitute it back into the expression for $\dddot{y}_2$, which becomes linear in α_{21} and allows for its unique recovery. Therefore, by combining the second and third derivatives of y_2, we obtain two independent algebraic constraints that together ensure the structural identifiability of both α_{21} and α_{22}.

2.4. Partially uncertain model structures

As discussed in the introductory Section 1.2, in order for the problem to be well-posed, we need to impose some *structure* to the uncertainty. In order to do this, we will partition the states into two components:

$$x = \begin{pmatrix} \xi \\ \zeta \end{pmatrix}$$

where $\xi \in \mathbb{R}^p$, $\zeta \in \mathbb{R}^q$, and $p + q = n$. The first block of variables will collect those mechanistic parts for which parameters are known.

We start with an example to illustrate the general idea:

$$\begin{aligned}
\dot{\xi}_1 &= f_1(x) + g_1(x)\\
\dot{\xi}_2 &= f_2(x) + g_2(x)\\
\dot{\xi}_3 &= f_3(x) + g_1(x) + g_3(x)\\
\dot{\zeta}_1 &= h_1(x) + g_2(x) + g_3(x)\\
\dot{\zeta}_2 &= h_2(x) + g_1(x) + 2g_2(x)
\end{aligned}$$

in which $p = 3$ and $q = 2$. We use lower case ξ_i, ζ_i for the coordinates of ξ, ζ. Here G is the vector with coordinates g_1, g_2, $g_1 + g_3$, $g_2 + g_3$, and $g_1 + 2g_2$ that represents how three "neural networks" affect the dynamics, and F has the coordinates f_1, f_2, f_3, h_1, h_2, where

$$h_1(x) = f_4(x, \pi),\ \ h_2(x) = f_5(x, \pi)$$

are the components of the mechanistic part of the model which contain the unknown parameters. As our focus is not on the values of the parameters π, we simply think of the h_i's as unknown functions that we wish to identify. Now, suppose that we have already obtained a model in which the coordinates of $x(t)$, that is, the functions $\xi_i(t)$ and $\zeta_i(t)$, have been found to fit given experimental data. There may be many sets of such functions x, obtained from different initial conditions or unmodeled factors. Since we know the functions $\xi_i(t)$, we may compute, at least in an ideal mathematical sense, their derivatives, and thus, subtracting the (known) function $f_i(x(t))$, we also know the functions $\gamma_1(t) = g_1(x(t))$ and $\gamma_2(t) = g_2(x(t))$. Similarly, from the knowledge of $\zeta_1(t)$ and $\zeta_2(t)$, we know $h_1(x(t)) = \dot{\zeta}_1(t) - (\gamma_2(t) + \gamma_3(t))$ and $h_2(x(t)) = \dot{\zeta}_2(t) - (\gamma_1(t) + 2\gamma_2(t))$. In summary, at least if the trajectories x are rich enough to explore the state space, we found the functions $h_1(x)$ and $h_2(x)$, which was our objective, We next make this discussion mathematically precise. We start by thinking of this example as follows. Let us introduce the following three matrices:

$$C = \begin{pmatrix} 1 & 0 & 1\\ 0 & 1 & 0\\ 0 & 0 & 1 \end{pmatrix},\ D = \begin{pmatrix} 0 & 1\\ 1 & 2\\ 1 & 0 \end{pmatrix},\ A = \begin{pmatrix} -1 & 1\\ 1 & 2\\ 1 & 0 \end{pmatrix}.$$

Observe that $D = CA$. This property explains why we could recover the functions h_i, as we will discuss next.

Definition. A *Partially Uncertain Model Structure* **(PUMS)** is a system structure as follows:

$$\begin{aligned}
\dot{\xi} &= \mathbf{f}(x) + C^\top \mathbf{g}(x)\\
\dot{\zeta} &= \mathbf{h}(x) + D^\top \mathbf{g}(x),
\end{aligned}$$

in which $x = (\xi, \zeta)^\top$. The functions **f**, **h**, and **g** map $\mathbb{R}^n$ into $\mathbb{R}^p$, $\mathbb{R}^q$, and $\mathbb{R}^r$ respectively, and $C \in \mathbb{R}^{r\times p}$, $D \in \mathbb{R}^{r\times q}$.

A PUMS structure is *identifiable* if, from knowledge of $x(t)$, we can recover **h** (i.e., unknown parameters in a known function). More formally, if the following property holds:
for all functions $\mathbf{g}$, $\widetilde{\mathbf{g}}$, $\mathbf{h}$, *and* $\widetilde{\mathbf{h}}$, *if:*

$$C^\top \mathbf{g}(x) = C^\top \widetilde{\mathbf{g}}(x) \ \text{ and } \ \mathbf{h}(x) + D^\top \mathbf{g}(x) = \widetilde{\mathbf{h}}(x) + D^\top \widetilde{\mathbf{g}}(x) \quad \forall\, x$$

then:

$$\mathbf{h}(x) = \widetilde{\mathbf{h}}(x) \quad \forall\, x\,.$$

Note that our example can be written as a PUMS, with $\mathbf{g} = (g_1, g_2, g_3)^\top$.

The interpretation of the identifiability definition is as follows. If we know that **f**, **g**, and **h** and **f**, $\widetilde{\mathbf{g}}$, and $\widetilde{\mathbf{h}}$, give the same trajectories $\xi(t)$ and

Table 2
Parameter values for the reduced 4D model used in simulations. Units are as appropriate (such as molarity per second (M/s) or Moles per liter per minute), but these are arbitrary parameters picked only in order to illustrate results.

Parameter	ω_{11}	ω_{21}	α_{11}	α_{21}	ω_{12}	ω_{22}	α_{12}	α_{22}	R_T
Value	1	1	1	3	3	7	3	5	20

$\dot{\zeta}(t)$, then necessarily the parameterized mechanistic parts $\mathbf{h}$ and $\widetilde{\mathbf{h}}$ must have been the same. In Theorem 1 (Section 3.3) we provide a necessary and sufficient condition. Intuitively the condition will say that from ξ we can obtain $C^\top \mathbf{g}$, because $\mathbf{f}(x)$ is known, and the condition given there then implies that $D^\top \mathbf{g}$ is known, so that $\mathbf{h}$ can be found from knowledge of this term as well as ζ.

3. Results

In this section, we explore our case study to demonstrate the implementation of the ePINNs approach for learning hidden mechanisms and predicting causal connections in the synthetic biology resource competition problem, and we provide a statement and proof of an identifiability result for PUMS.

We aim to explore the following research questions through this case study:

1. *System observability*: Using only the selected output measurements, can the trajectories of all states be accurately predicted?
2. *Parameter identification*: Can unknown parameters (chemical reaction rates, in this example) be inferred effectively?
3. *Missing dynamics prediction*: Can the "black-box" hidden mechanism (in this example, related to resource competition) be uncovered?
4. *Comparing scenarios:* Does even partial knowledge improve fits, compared to a purely "black box" approach?

We will evaluate several scenarios with varying levels of system knowledge for the described case study.

3.1. 4D model

In this section, we use the 4D model and the parameters shown in Table 2 to generate the synthetic data used for training ePINNs. To simulate experimental uncertainty, we introduced additive Gaussian noise to the synthetic data. For each observed state variable x, the noise term $\eta(t)$ was sampled from a normal distribution $\mathcal{N}(0, \sigma^2)$, where σ was set to 5% of the variability of x over the sampling period, denoted by $\mathrm{Var}(x)$. The resulting noisy state variables are given by $\tilde{x}(t) = x(t) + \eta(t)$, where $\eta(t) \sim \mathcal{N}(0, (0.05 \cdot \mathrm{Var}(x))^2)$. This ensures that the noise magnitude is proportional to the natural variability of each observed state over time. The noisy state trajectories $\tilde{x}(t)$ were used for training.

We then evaluate the performance of the ePINNs approach for the defined resource competition problem with the same 4D dynamics under the following scenarios.

3.1.1. Impact of full mechanistic knowledge on estimation and prediction with partial observations

To make the problem interesting (and more biologically realistic), we use only the partial observations represented by the output variable $y = (F, M + M_I)^\top$ instead of the four state variables. One might reasonably ask whether the use of restricted information, meaning that we need to solve a simultaneous systems identification and observability problem, is the reason that some approaches might perform better than others. To control for this effect, we start by examining how the availability of full knowledge about system dynamics influences the estimation and prediction of behavior when only partial observations are available. In this case, the unknown mechanism and the blue parameters in our 4D system are assumed to be known. Our goal is to evaluate whether ePINNs can successfully reconstruct all state variables using only partial, noisy measurements (question (1)). This tests the method's ability to infer unmeasured dynamics and assess observability. We assume that only data from total mRNA $(M + M_I)$ and the mRNA-ribosome complex (F) are available.

Fig. 3 illustrates the results obtained from a "physically **un**informed" neural network (set the mechanistic loss weight $w^{\mathrm{ODE}} = 0$), where predictions are represented by solid green lines and labeled as "NNs." By comparison, Fig. 4 shows the results from a physically informed neural network ($w^{\mathrm{ODE}} = 0.5$), where predictions are represented by solid green lines and labeled as "ePINNs." Both approaches provide accurate estimations for the observed data (F and $M + M_I$).

Although both approaches utilize the available data, it is evident from Figs. 5 and 6 that the physically uninformed neural network provides inaccurate estimations for the state components R, M, and M_I, for which no direct data is available. However, as shown in Fig. 6, the physically informed network effectively leverages the system's physics knowledge to achieve accurate estimations for all states, even in the absence of direct measurements of all states (question (4)).

We note that both training processes (for the informed and uninformed cases discussed above) were conducted under identical conditions, network structures, and observed noisy data to ensure a fair comparison. In both cases, a learning rate of 0.01 was employed, with the first neural network consisting of 4 fully connected hidden layers, each containing 64 neurons. The "softplus" activation function was chosen for both scenarios. The training phase spanned from 0 to 15, while the testing phase extended from 15 to 20 to evaluate the neural networks' accuracy in predicting state behaviors beyond the training period. The synthetic data were sampled non-uniformly: at intervals of 0.01 for the initial transient phase ($t \le 2.5$) and 0.05 thereafter. This sampling strategy allows finer resolution during fast transients and coarser resolution during slower dynamics. An epoch count of $180k$ was selected uniformly across all cases.

3.1.2. Partially observed and partially known system identification

This scenario represents the most realistic case, where both full state measurements and system knowledge are incomplete. Missing data and unknown interactions challenge the ePINNs framework to simultaneously infer the unknown parameters α_{21} and α_{22} (question (2)), as well as the missing terms in the gray box (question (3)).

Training data is assumed to be available only for the total mRNA $(M + M_I)$ and the mRNA-ribosome complex (F), leaving the system partially observed. The first neural network $(\mathcal{X})$ estimates the trajectory as a function of time , while the second neural network $(\mathcal{G})$ models the unknown ribosome competition term within the gray box in our 4D system. The trained model incorporates the dynamics of $\frac{dR}{dt}$ and $\frac{dM_I}{dt}$ as follows:

$$\begin{aligned}\frac{d\hat{R}}{dt} &= -w_{11}\hat{M}\hat{R} + w_{12}\hat{F} + \mathcal{G}(\hat{x}, \theta_2),\\ \frac{d\hat{M}_I}{dt} &= \alpha_{21} - \alpha_{22}\hat{M}_I + \mathcal{G}(\hat{x}, \theta_2),\end{aligned} \tag{3}$$

where $\hat{R}$, $\hat{M}$, $\hat{F}$, and $\hat{M}_I$ are the first neural network output states corresponding to R, M, F, and M_I, respectively.

The learning patterns for the unknown parameters α_{21} and α_{22} are presented in Fig. 7, showing the convergence of ePINNs predictions (solid green lines) to their true values (dashed red lines) using noisy observations from F and $M + M_I$.

To evaluate missing dynamics prediction, Fig. 8 illustrates how ePINNs reconstruct the hidden competition effect $G(x)$. Despite only partial data availability, the model successfully learns the underlying mechanism, with the predicted dynamics (solid green line) closely matching the ground truth (red dots).

Finally, Fig. 9 provides a comprehensive evaluation of system observability, parameter identification, and missing dynamics prediction. The

Table 3
Parameter values for the 7D model used in simulations. Units are as appropriate (such as molarity per second (M/s) or Moles per liter per minute), but these are arbitrary parameters picked only in order to illustrate results.

Parameter	ω_{11}	ω_{21}	α_{11}	α_{21}	k_{11}	k_{21}	β_1	Q_T	U_T
Value	1	1	1	3	1	1	2	18	15
Parameter	ω_{12}	ω_{22}	α_{12}	α_{22}	k_{12}	k_{22}	β_2	R_T	$U_{I,T}$
Value	3	7	3	5	8	12	1	20	26

ePINNs model successfully reconstructs all state variables (R, M, F, M_I) by leveraging available data and mechanistic constraints, highlighting its effectiveness in hybrid modeling scenarios. The training process for this example was conducted with a learning rate of 0.02. The first neural network consisted of 4 fully connected hidden layers, each containing 64 neurons, while the second neural network had 2 layers with 64 neurons each. The “softplus" activation function was used for both networks. An epoch count of 420k was applied uniformly across all cases.

3.2. 7D model

In this section, we use the 7D model and the parameters shown in Table 3 to generate the synthetic data used for training ePINNs. To simulate experimental uncertainty, we use the same method explained for the 4D model. We use Y, M, and M_I as outputs for the training purposes.

For this example, training was performed with a scheduled learning rate. The architecture consisted of two neural networks: the first network featured four fully connected hidden layers with 64 neurons per layer, while the second comprised two layers each with 64 neurons. The “softplus" activation function was employed throughout. This model was trained for 280,000 epochs.

Fig. 10 shows the evolution of the learned parameters α_{21}, α_{22}, and k_{21}. The ePINNs estimates closely track the true values, indicating accurate parameter identification from limited observations.

3.3. A result about PUMS identifiability

Theorem 1. *A PUMS is identifiable if and only if col $D \subseteq$ col C*

Here, col M denotes the column span of a matrix M. Notice that asking that col $D \subseteq$ col C is equivalent to asking that there exists a matrix A such that $D = CA$. So our example is an identifiable structure in this sense.

Proof. Suppose that $D = CA$. Take functions introduced earlier in the definition of PUMS, so we have that

$$C^\top(\mathbf{g}(x) - \widetilde{\mathbf{g}}(x)) = 0\,.$$

Since $D^\top = A^\top C^\top$, it follows that

$$D^\top(\mathbf{g}(x) - \widetilde{\mathbf{g}}(x)) = A^\top C^\top(\mathbf{g}(x) - \widetilde{\mathbf{g}}(x)) = 0$$

and substituting this into

$$(\mathbf{h}(x) - \widetilde{\mathbf{h}}(x)) + D^\top(\mathbf{g}(x) - \widetilde{\mathbf{g}}(x)) = 0$$

we conclude that $\mathbf{h}(x) - \widetilde{\mathbf{h}}(x) = 0$.

To prove the converse implication, suppose that col D is not included in col C. This means that there is a column of D which is not in the span of the columns of C. Denote the columns of D as $\mathbf{d}_i = (d_{1i}, \dots, d_{ri})^\top$, for $i = 1, \dots, q$. Suppose that $\mathbf{d}_j \notin$ col C for some j. Then there exists a vector $v \in \mathbb{R}^r$ in $C^\perp$ (so that we have $C^\top v = 0$) such that $v^\top \mathbf{d}_j \neq 0$. Now pick any $\mathbf{g}$ and $\mathbf{h}$ (for example $\mathbf{g} = 0$ and $\mathbf{h} = 0$). We need to find functions $\widetilde{\mathbf{g}}$ and $\widetilde{\mathbf{h}}$ such that (omitting arguments for simplicity): (1) $C^\top \mathbf{g} = C^\top \widetilde{\mathbf{g}}$ and (2) $\mathbf{h} + D^\top \mathbf{g} = \widetilde{\mathbf{h}} + D^\top \widetilde{\mathbf{g}}$, but $\mathbf{h} \neq \widetilde{\mathbf{h}}$. Define $\widetilde{\mathbf{g}} := \mathbf{g} + v$ (as a constant function) and $\widetilde{\mathbf{h}} := \mathbf{h} - D^\top v$. Note that $D^\top v \neq 0$ since $v^\top \mathbf{d}_j \neq 0$, so $\mathbf{h} \neq \widetilde{\mathbf{h}}$. We have that $C^\top(\mathbf{g} - \widetilde{\mathbf{g}}) = -C^\top v = 0$, so (1) holds. From

$$\widetilde{\mathbf{h}} + D^\top \widetilde{\mathbf{g}} = (\mathbf{h} - D^\top v) + D^\top(\mathbf{g} + v) = \mathbf{h} + D^\top \mathbf{g},$$

we have that (2) holds as well. This completes the proof. □

4. Discussion

In this work, we introduced the embedded Physics-Informed Neural Networks (ePINNs) framework to integrate mechanistic modeling with data-driven learning for systems with incomplete knowledge. By combining known system dynamics with neural network-based inference, ePINNs address challenges in modeling biological systems where reaction mechanisms or system parameters are partially unknown.

Applying ePINNs to the study of resource competition in synthetic biology, we demonstrated, through a particular example, their potential ability to infer unmeasured reaction rates from noisy and incomplete observations, predict system behavior beyond observed data using governing ODE constraints, and identify missing dynamics without prior explicit knowledge. Our results showed that physics-informed learning improves estimation accuracy compared to purely data-driven models, particularly for unobserved states. The framework effectively leverages available mechanistic insights while learning unknown interactions, making it adaptable to a broad range of complex biological problems.

While related to Universal PINNs (UPINNs), our work advances this direction by introducing the PUMS formalism with an identifiability result for partial observability and by demonstrating its utility on a biologically motivated resource-competition problem. These contributions extended the scope of hybrid modeling, emphasizing identifiability as a central consideration for applying such methods to complex biological systems. This is especially relevant when parameters are unknown, since it is then impossible to simply “subtract" the effect of known mechanistic components and fit the“residual" error.

In ongoing work, we are evaluating the approach on the more complete resource competition model as well as its performance with other parameter values.

Acknowledgments

This work was supported in part by grants AFOSR FA9550-22-1-0316 and ONR N00014-21-1-2431, as well as funding from the Institute for Experiential AI at Northeastern University.

Software

Programs used to generate figures in this report are placed in: *https://github.com/sontaglab/epinn*.

CRediT authorship contribution statement

Atefe Darabi: Writing - review & editing, Writing - original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Zheming An:** Writing - original draft, Software, Investigation, Formal analysis, Conceptualization; **Muhammad Ali Al-Radhawi:** Writing - original draft, Methodology, Conceptualization; **William Cho:** Software; **Milad Siami:** Writing - review & editing, Supervision, Funding acquisition; **Eduardo D. Sontag:** Writing - review & editing, Writing - original draft, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[2] M.I. Jordan, T.M. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.

[3] A.A. Nielsen, C.A. Voigt, Machine learning in synthetic biology: new design paradigms for genetic circuits, Curr. Opin. Chem. Biol. 34 (2016) 135–144.

[4] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (5) (2019) 206–215.

[5] D. Del Vecchio, Y. Qian, R. Murray, E. Sontag, Future systems and control research in synthetic biology, Annu. Rev. Control 45 (2018) 5–17.

[6] S.H. Strogatz, Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering, CRC Press, 2001.

[7] D.C. Psichogios, L.H. Ungar, A hybrid neural network-first principles approach to process modeling, AlChE J. 38 (10) (1992) 1499–1511.

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

[9] M. Thompson, M. Kramer, Modeling chemical processes using prior knowledge and neural networks, AlChE J. 40 (1994) 1328–1340.

[10] M. von Stosch, R. Oliveira, J. Peres, S. Feyo de Azevedo, Hybrid semi-parametric modeling in process systems engineering: past, present and future, Comput. Chem. Eng. 60 (2014) 86–101.

[11] M. von Stosch, J. Peres, S.F. de Azevedo, R. Oliveira, Modelling biochemical networks with intrinsic time delays: a hybrid semi-parametric approach, BMC Syst. Biol. 4 (2010) 131.

[12] B. Engelhardt, H. Frőhlich, M. Kschischo, Learning (from) the errors of a systems biology model, Sci. Rep. 6 (1) (2016) 20772.

[13] D. Lee, A. Jayaraman, J. Kwon, Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling, PLoS Comput. Biol. 16 (2020) 1–31.

[14] A. Procopio, G. Cesarelli, L. Donisi, A. Merola, F. Amato, C. Cosentino, Combined mechanistic modeling and machine-learning approaches in systems biology - a systematic literature review, Comput. Methods Programs Biomed. 240 (2023) 107681.

[15] D. Del Vecchio, A. Ninfa, E. Sontag, Modular cell biology: retroactivity and insulation, Mol. Syst. Biol. 4 (2008) 161.

[16] E. Yeung, A.J. Dy, K.B. Martin, A.H. Ng, D.D. Vecchio, J.L. Beck, J.J. Collins, R.M. Murray, Biophysical constraints arising from compositional context in synthetic gene networks, Cell Syst. 5 (1) (2017) 11–24.e12.

[17] M.A. Al-Radhawi, K. Manoj, D. Jatkar, A. Duvall, D. Del Vecchio, E. Sontag, Competition for binding targets results in paradoxical effects for simultaneous activator and repressor action, in: Proc. 63rd IEEE Conference on Decision and Control (CDC), (2024).

[18] M. Al-Radhawi, S. Tripathi, Y. Zhang, E. Sontag, H. Levine, Epigenetic factor competition reshapes the EMT landscape, Proc. Natl. Acad. Sci. U.S.A. 119 (2022) e2210844119. https://doi.org/10.1073/pnas.2210844119.

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

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

[21] M. Dissanayake, N. Phan-Thien, Neural-network-based approximations for solving partial differential equations, Commun. Numer. Methods Eng. 10 (3) (1994) 195–201.

[22] B. Eastman, L. Podina, M. Kohandel, A pinn approach to symbolic differential operator discovery with sparse data, in: The Symbiosis of Deep Learning and Differential Equations II, (2022).

[23] L. Podina, B. Eastman, M. Kohandel, Universal physics-informed neural networks: symbolic differential operator discovery with sparse data, in: International Conference on Machine Learning, PMLR, (2023), pp. 27948–27956.

[24] L. Podina, D. Darooneh, J. Grewal, M. Kohandel, Enhancing symbolic regression and universal physics-informed neural networks with dimensional analysis, arXiv:2411.15919 (2024).

[25] L. Podina, A. Ghodsi, M. Kohandel, Learning chemotherapy drug action via universal physics-informed neural networks, Pharm. Res. 42 (4) (2025) 593–612.

[26] Z. An, W. Cho, M. Sadeghi, E.D. Sontag, Physics informed neural network for parameter and hidden dynamics inference: A metronomic chemotherapy model reduction problem, Poster presented at Northeastern University, 2022.

[27] Z. An, M.A. Al-Radhawi, W. Cho, E.D. Sontag, Embedded physics informed neural networks (ePINN) for parameter and hidden dynamics inference: Identifying resource competition phenotypes in synthetic biochemical circuits, Poster presented at Northeastern University, 2023.

[28] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2018) 1–43.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS 2017 Workshop on Autodiff, (2017), pp. 1–4.

[30] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 (2014).

[31] M. Al-Radhawi, D. Del Vecchio, E. Sontag, Identifying competition phenotypes in synthetic biochemical circuits, IEEE Control Syst. Lett. 7 (2023) 211–216.

[32] A. Raveh, M. Margaliot, E. Sontag, T. Tuller, A model for competition for ribosomes in the cell, Proc. R. Soc. Interface 13 (2016) 2015.1062.

[33] J. Miller, M. Al-Radhawi, E. Sontag, Mediating ribosomal competition by splitting pools, IEEE Control Syst. Lett. 5 (2020) 1555–1560.

[34] E. Sontag, Mathematical Control Theory. Deterministic Finite-Dimensional Systems, Springer-Verlag, 1998.

[35] A. Martinelli, Extension of the observability rank condition to nonlinear systems driven by unknown inputs, in: Proceedings of the 23rd Mediterranean Conference on Control and Automation (MED), (2015), pp. 589–595.

[36] K. Maes, M.N. Chatzis, G. Lombaert, Observability of nonlinear systems with unmeasured inputs, Mech. Syst. Signal Process. 130 (2019) 378–394.

[37] X. Shi, M. Chatzis, An efficient algorithm to test the observability of rational nonlinear systems with unmeasured inputs, Mech. Syst. Signal Process. 165 (2022) 108345.

[38] S. Díaz-Seoane, X. Rey Barreiro, A.F. Villaverde, Strike-goldd 4.0: user-friendly, efficient analysis of structural identifiability and observability, Bioinformatics 39 (1) (2022) btac748. https://doi.org/10.1093/bioinformatics/btac748.