COMBINING MODEL-BASED AND DATA-DRIVEN MODELS: AN APPLICATION TO SYNTHETIC BIOLOGY RESOURCE COMPETITION

ATEFE DARABI*, ZHEMING AN*, MUHAMMAD ALI AL-RADHAWI, WILLIAM CHO, MILAD SIAMI, AND EDUARDO D. SONTAG

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.

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, 12, 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 2.3. 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, resource sharing, genetic context effects, growth rate feedback, and off-target interactions can disrupt intended circuit functions [15, 16, 5] 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.

In this work, we extend the PINNs idea, and develop the concept of *embedded PINNs (ePINNs)*, originally proposed in [22, 23], which is a novel framework composed of two loss-sharing neural networks designed to improve predictions of unknown dynamics from quantitative time-series data. 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 $y = (y_1, \ldots, y_l)$ 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:

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

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

The structure of ePINNs is depicted in Figure 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 $G(x)$.

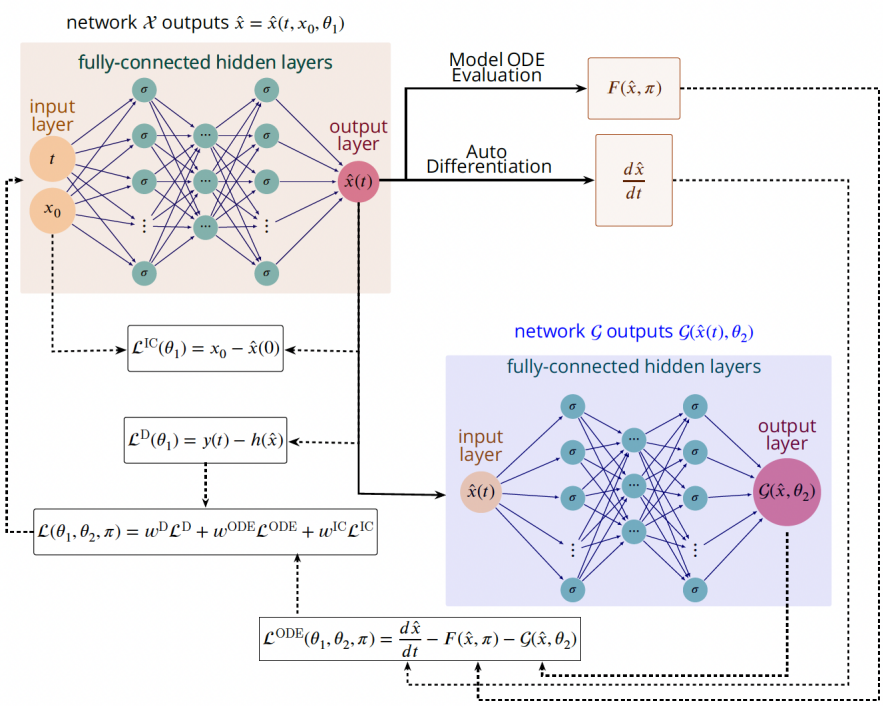

FIGURE 1. **ePINNs architecture.** ePINNs structure is composed of two embedded neural networks. The first one, $\mathcal{X}$, has weights θ_1, postulates a trajectory $\hat{x}(t)$ as a function of time and initial conditions $x(0)$. The second one, $\mathcal{G}$, has weights θ_2 and takes as its input 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.

2.1.1. Overall loss function structure

The two networks do not interact directly, but they both contribute to the ODE loss function $\mathcal{L}^{\mathrm{ODE}}(\theta_1,\theta_2,\pi)$, 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*, $\mathcal{L}^{\text{D}}(\theta_1)$: Ensures that state predictions from $\mathcal{X}$ match experimental observations.
(2) *ODE constraint*, $\mathcal{L}^{\text{ODE}}(\theta_1, \theta_2, \pi)$: Ensures that derivative predictions $\hat{x}$, combined with unknown dynamics from $\mathcal{G}$, satisfy the ODE system.
(3) *Initial condition constraint*, $\mathcal{L}^{\text{IC}}(\theta_1)$: 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 $t = (t_1, \ldots, t_N)$. 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 reconstruct π 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.

We introduce coefficients, denoted by w^{D}, w^{ODE}, and w^{IC}, respectively, to balance the relative contributions of $\mathcal{L}^{\text{D}}$, $\mathcal{L}^{\text{ODE}}$, and $\mathcal{L}^{\text{IC}}$ into an overall loss function:

$$\mathcal{L}(\theta_1, \theta_2, \pi) \;:=\; w^{\text{D}}\mathcal{L}^{\text{D}}(\theta_1) + w^{\text{ODE}}\mathcal{L}^{\text{ODE}}(\theta_1, \theta_2, \pi) + w^{\text{IC}}\mathcal{L}^{\text{IC}}(\theta_1)\,.$$

The explicit forms of losses are defined as follows:

$$\begin{aligned}
\mathcal{L}^{\text{D}}(\theta_1) \;&:=\; \frac{1}{N}\sum_{j=1}^{N} \left\| y(t_j) - h(\hat{x}(t_j)) \right\|^2, \\
\mathcal{L}^{\text{ODE}}(\theta_1, \theta_2, \pi) \;&:=\; \frac{1}{N}\sum_{j=1}^{N} \left\| \frac{d\hat{x}}{dt}(t_j) - F(\hat{x}(t_j), \pi) - \mathcal{G}(\hat{x}(t_j), \theta_2) \right\|^2, \\
\mathcal{L}^{\text{IC}}(\theta_1) \;&:=\; \left\| x(0) - x_0 \right\|^2.
\end{aligned}$$

The loss functions are also shown in Figure 1 The loss components $\mathcal{L}^{\text{D}}(\theta_1)$ and $\mathcal{L}^{\text{IC}}(\theta_1)$ deal with mismatches between model predictions and observations y. On the other hand, the component $\mathcal{L}^{\text{ODE}}(\theta_1, \theta_2, \pi)$ is not directly dependent on data and only relies upon ODE evaluations and automatic differentiation (AD), which computes $\frac{d\hat{x}}{dt}$ [24, 25]. 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 [26], to update the parameters θ_1, θ_2, and π simultaneously. The optimized parameters $(\theta_1^*, \theta_2^*, \pi^*)$ are obtained by minimizing the shared loss function $\mathcal{L}(\theta_1, \theta_2, \pi)$, as follows:

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

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 Figure 2.

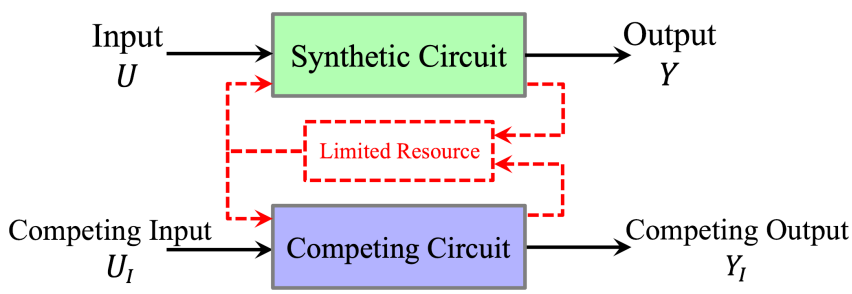

FIGURE 2. A synthetic circuit with one competing circuit.

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^f. 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 subscripts I to denote components of this other circuit; it has input U_I^f 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:

$$\mathrm{U^f} + \mathrm{Q} \underset{k_{12}}{\overset{k_{11}}{\rightleftharpoons}} \mathrm{E} \xrightarrow{\alpha_{11}} \mathrm{U^f} + \mathrm{Q} + \mathrm{M}\,, \ \mathrm{M} \xrightarrow{\alpha_{12}} \varnothing$$

$$\mathrm{M} + \mathrm{R} \underset{\omega_{12}}{\overset{\omega_{11}}{\rightleftharpoons}} \mathrm{F} \xrightarrow{\beta_1} \mathrm{M} + \mathrm{R} + \mathrm{Y}\,, \ \mathrm{Y} \xrightarrow{\beta_2} \varnothing$$

$$\mathrm{U_I^f} + \mathrm{Q} \underset{k_{22}}{\overset{k_{21}}{\rightleftharpoons}} \mathrm{E_I} \xrightarrow{\alpha_{21}} \mathrm{U_I^f} + \mathrm{Q} + \mathrm{M_I}\,, \ \mathrm{M_I} \xrightarrow{\alpha_{22}} \varnothing$$

$$\mathrm{M_I} + \mathrm{R} \underset{\omega_{22}}{\overset{\omega_{21}}{\rightleftharpoons}} \mathrm{F_I} \xrightarrow{\beta_1} \mathrm{M_I} + \mathrm{R} + \mathrm{Y_I}\,, \ \mathrm{Y_I} \xrightarrow{\beta_2} \varnothing.$$

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^f, E, M, F, Y for the synthetic circuit, and $U_I^f, E_I, M_I, F_I, Y_I$ 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.

Synthetic Circuit	**Competing Circuit**	**Resources**
U^f: free promoter	U_I^f: 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	

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

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

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

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

$$\begin{aligned} U^f + E &\equiv U^f_T \\ U^f_I + E_I &\equiv U^f_{I,T} \\ R + F + F_I &\equiv R_T \\ Q + E + E_I &\equiv Q_T \end{aligned}$$

(for example, the time derivative of $U^f(t) + E(t)$ 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 $Q(0) = Q_T$, $R(0) = R_T$, $U^f(0) = U^f_T$, $U^f_I(0) = U^f_{I,T}$, with the remaining variables set to zero.

RNAP and ribosome competition give rise to different behaviors, studied for example in [27]. Let us focus in 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 $Q(t) \equiv Q_T$. What this means is that we will drop the differential equation for $Q(t)$ and replace Q by Q_T in all the equations. To simplify further, under an assumption of rapid binding and unbinding of RNAP to promoters, we assume an equilibrium value for the free promoters U^f and U^f_I. Note that the output proteins Y and Y_I do not directly affect the reaction chains defined in the system, so we will neglect them in a simplified model. Applying the first three conservation laws, and incorporating these assumptions, we reduce the 12-dimensional system to the following simplified 4-dimensional (4D) model:

$$\begin{aligned} \dot{R} &= -w_{11}MR + w_{12}F \boxed{-w_{21}M_IR + w_{22}(R_T - R - F)}, \\ \dot{M} &= \alpha_{11} - \alpha_{12}M - w_{11}MR + w_{12}F, \\ \dot{F} &= w_{11}MR - w_{12}F, \\ \dot{M}_I &= \alpha_{21} - \alpha_{22}M_I \boxed{-w_{21}M_IR + w_{22}(R_T - R - F)}. \end{aligned}$$

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). Observe that we have used a conservation law to write $F_I = R_T - (R + F)$. 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). The remaining parameters are supposed to be 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 $x = (R, M, F, M_I)^\top$.) Of course, if we knew that the interaction between M_I and R has this particular form, the problem would be reduced to the parameter-fitting problem of finding w_{21} and w_{22} in addition to α_{21} and α_{22}. 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 [28] and in particular the splitting of the ribosome pool by use of orthogonal ribosomes [29].

We will use this 4D system in order to numerically explore how partial mechanistic knowledge can improve data fits using ePINNs. The approach will be to numerically generate data from a "ground truth" system and to attempt to use ePINNs to fit this data. The system generating this data will the parameters shown in Table 2.

Parameter	ω_{11}	ω_{21}	α_{11}	α_{21}	β_1	Q_T
Value	1	1	1	3	2	18
Parameter	ω_{12}	ω_{22}	α_{12}	α_{22}	β_2	R_T
Value	3	7	3	5	1	20

TABLE 2. Parameter values for the reduced 4D model used in simulations. Units are as appropriate (concentrations/second, etc), but these are arbitrary parameters picked only in order to illustrate results.

By focusing on the reduced 4D system, we aim to highlight the key concepts more clearly while avoiding unnecessary complexity. In a follow-up work, we will extend the analysis to the full system.

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 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.

2.3. **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 $x(t)$ can recover **h** (i.e., parameters). 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 $\dot{\xi}(t)$ and $\dot{\zeta}(t)$, then necessarily the parameterized mechanistic parts **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 **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.

Synthetic Data Generation. 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 $\text{Var}(x)$. The resulting noisy state variables are given by $\tilde{x}(t) = x(t) + \eta(t), \quad \text{where} \quad \eta(t) \sim \mathcal{N}(0, (0.05 \cdot \text{Var}(x))^2)$. 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. 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.

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

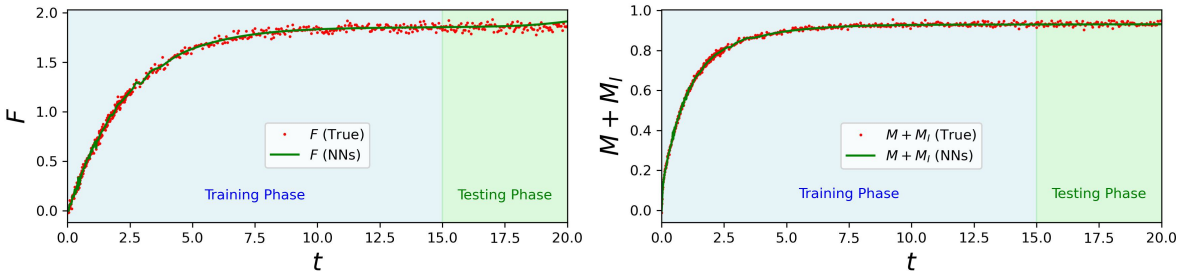

FIGURE 3. **Physically uninformed neural network ($w^{\text{ODE}} = 0$) results for observables.** Predictions for state variables R, $M + M_I$ (solid green lines, labeled "NNs") are shown using noisy data from F and $M + M_I$ (red dots, labeled "True"). The physically uninformed network provides an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

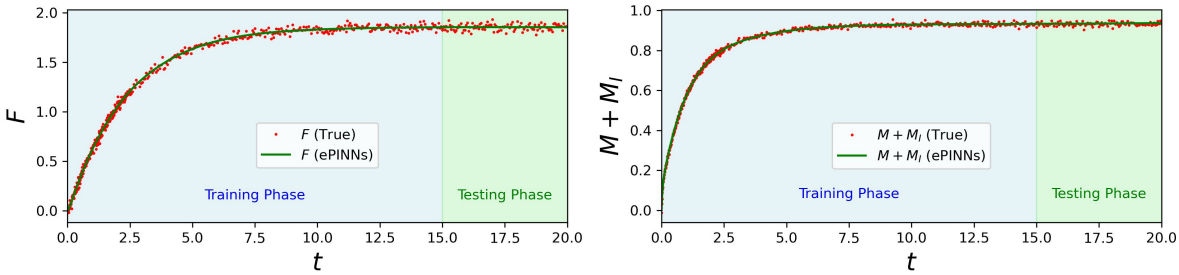

FIGURE 4. **Physically informed neural network ($w^{\text{ODE}} = 0.5$) results for observables.** Predictions for state variables R and $M + M_I$ (solid green lines, labeled "ePINNs") are shown using noisy data from F and $M + M_I$ (red dots, labeled "True"). The physically informed network provides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

Although both approaches utilize the available data, it is evident from Figures 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 Figure 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. An epoch count of $180k$ was selected uniformly across all cases.

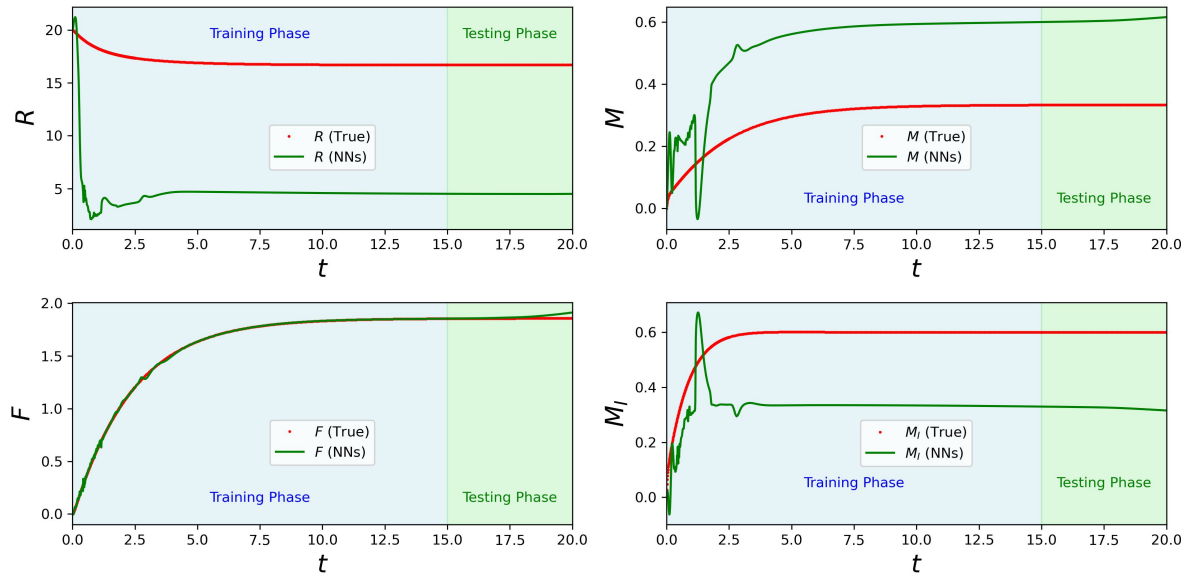

FIGURE 5. **Physically uninformed neural network ($w^{\text{ODE}} = 0$) predictions for all states.** Predictions for state variables R, M, and M_I (solid green lines, labeled "NNs") are shown using noisy data from F and $M + M_I$ (red dots, labeled "True"). The physically uninformed network fails to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

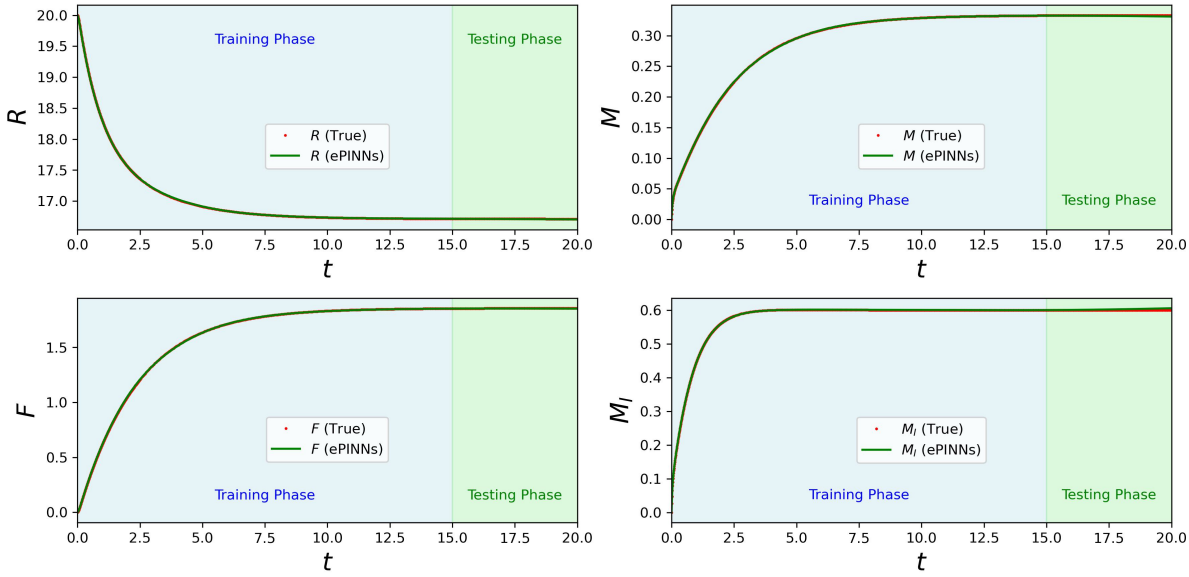

FIGURE 6. **Physically informed neural network ($w^{\text{ODE}} = 0.5$) predictions for all states.** Predictions for state variables R, M, and M_I (solid green lines, labeled "ePINNs") are shown using noisy data from F and $M+M_I$ (red dots, labeled "True"). The physically informed network accurately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

3.2. **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 α_{21} and α_{22}, 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 $\hat{R}$ and $\hat{M}_I$ are the first neural network output states corresponding to R and M_I, respectively.

The learning patterns for the unknown parameters α_{21} and α_{22} are presented in Figure 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, Figure 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, Figure 9 provides a comprehensive evaluation of system observability, parameter identification, and missing dynamics prediction. The 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 400k was applied uniformly across all cases.

3.3. **A result about PUMS identifiability**

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

Here, $\operatorname{col} M$ denotes the column span of a matrix M. Notice that asking that $\operatorname{col} D \subseteq \operatorname{col} C$ 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 as above, so we have that

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

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

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

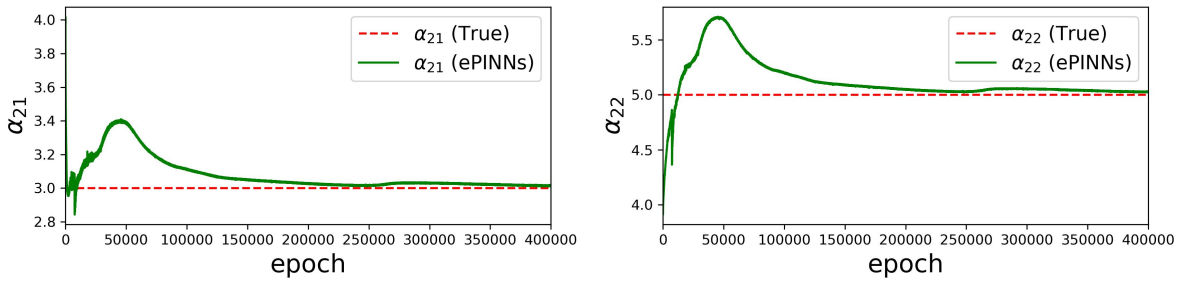

FIGURE 7. **Learning patterns for unknown parameters α_{21} and α_{22} with partial data and missing dynamics.** The solid green lines (labeled "ePINNs") represent the estimated values, while the dashed red lines indicate the true values. The training process uses noisy observations from F and $M + M_I$, demonstrating ePINNs' ability to identify unknown parameters.

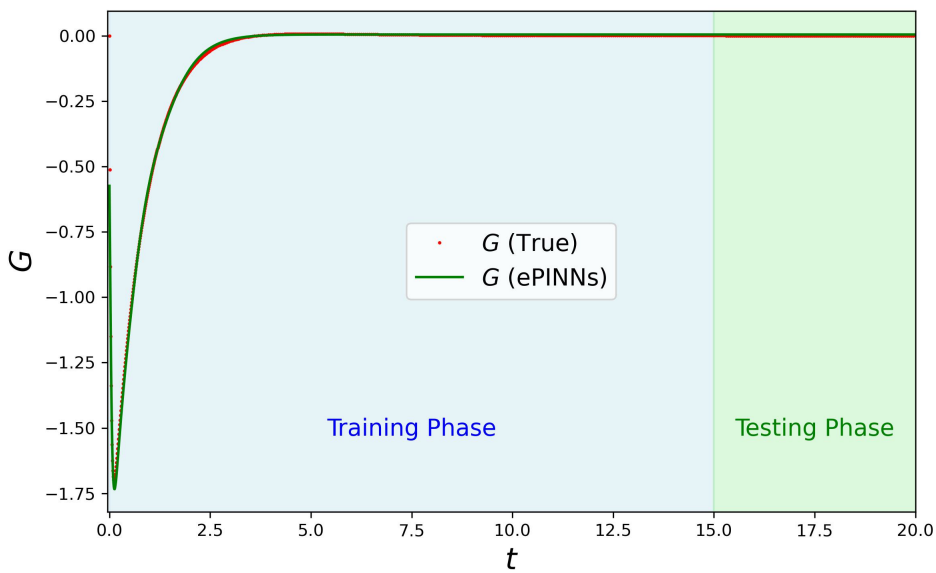

FIGURE 8. **ePINNs prediction of missing dynamics $G(x)$ with partial data.** The solid green line (labeled "ePINNs") represents the model's prediction for the unknown dynamics, while red dots (labeled "True") indicate ground truth values. The training uses noisy observations from F and $M + M_I$, demonstrating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

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 $\mathbf{d}_i = (d_{1i}, \ldots, d_{ri})^\top$, for $i = 1, \ldots, q$. Suppose that $\mathbf{d}_j \notin \operatorname{col} C$ for some j. Then there exists a vector $\nu \in \mathbb{R}^r$ in $C^\perp$ (so that we have $C^\top \nu = 0$) such that $\nu^\top \mathbf{d}_j \neq 0$. Now pick any $\mathbf{g}$

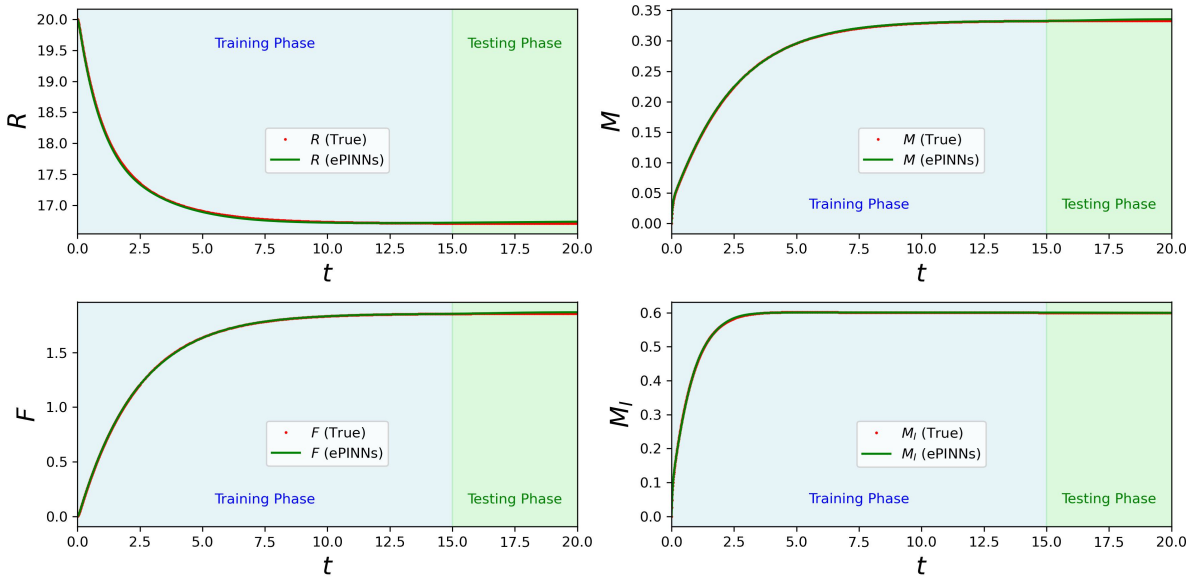

FIGURE 9. **Predictions for all state variables with partial data and missing dynamics.** The solid green lines (labeled "ePINNs") show the predicted states R, M, F, and M_I, while red dots (labeled "True") represent the observed data for F and $M + M_I$. The results illustrate the effectiveness of ePINNs in reconstructing unobserved states using physics-informed constraints.

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 $\widetilde{\mathbf{g}} := \mathbf{g} + \nu$ (as a constant function) and $\widetilde{\mathbf{h}} := \mathbf{h} - D^\top \nu$. Note that $D^\top \nu \neq 0$ since $\nu^\top \mathbf{d}_j \neq 0$, so $\mathbf{h} \neq \widetilde{\mathbf{h}}$. We have that $C^\top(\mathbf{g} - \widetilde{\mathbf{g}}) = -C^\top \nu = 0$, so (1) holds. From

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

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.

In ongoing work, we are evaluating the approach on the more complete resource competition model (as opposed to the reduced 4D model) and 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.

SOFTWARE

Programs used to generate figures in this report are placed in:

https://github.com/sontaglab/epinn

REFERENCES

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

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

[3] A. A. Nielsen and C. A. Voigt. Machine learning in synthetic biology: new design paradigms for genetic circuits. *Current Opinion in Chemical Biology*, 34:135–144, 2016.

[4] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[5] D. Del Vecchio, Y. Qian, R. Murray, and E. Sontag. Future systems and control research in synthetic biology. *Annual Reviews in Control*, 45:5–17, 2018.

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

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

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[9] M. Thompson and M. Kramer. Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, 40:1328–1340, 1994.

[10] M. von Stosch, R. Oliveira, J. Peres, and S. Feyo de Azevedo. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering*, 60:86–101, 2014. ISSN 0098-1354.

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

[12] B. Engelhardt, H. Frőhlich, and M. Kschischo. Learning (from) the errors of a systems biology model. *Scientific Reports*, 6(1):20772, Feb 2016. ISSN 2045-2322.

[13] D. Lee, A. Jayaraman, and J. Kwon. Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling. *PLOS Computational Biology*, 16:1–31, 12 2020.

[14] A. Procopio, G. Cesarelli, L. Donisi, A. Merola, F. Amato, and C. Cosentino. Combined mechanistic modeling and machine-learning approaches in systems biology – a systematic literature review. *Computer Methods and Programs in Biomedicine*, 240:107681, 2023. ISSN 0169-2607.

[15] D. Del Vecchio, A. Ninfa, and E. Sontag. Modular cell biology: Retroactivity and insulation. *Molecular Systems Biology*, 4:161, 2008.

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

[17] M. A. Al-Radhawi, K. Manoj, D. Jatkar, A. Duvall, D. Del Vecchio, and 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. To appear. Preprint in arXiv.

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

[19] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[20] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[21] M. Dissanayake and N. Phan-Thien. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 10 (3):195–201, 1994.

[22] Z. An, W. Cho, M. Sadeghi, and E. D. Sontag. Physics informed neural network for parameter and hidden dynamics inference: A metronomic chemotherapy model reduction problem, 2022. Poster presented at Experiential AI Institute Postdoc Mid-Year Progression Meeting, The Roux Institute at Northeastern University, Portland, ME, May 23, 2022.

[23] Z. An, M. A. Al-Radhawi, W. Cho, and E. D. Sontag. Embedded physics informed neural networks (ePINN) for parameter and hidden dynamics inference: Identifying resource competition phenotypes in synthetic biochemical circuits, 2023. Poster presented at Northeastern University Synthetic Biology Workshop, Boston, MA, April 20, 2023.

[24] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18:1–43, 2018.

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, pages 1–4, 2017.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] M. Al-Radhawi, D. Del Vecchio, and E. Sontag. Identifying competition phenotypes in synthetic biochemical circuits. *IEEE Control Systems Letters*, 7:211–216, 2023. (Online published in 2022; in print 2023.).

[28] A. Raveh, M. Margaliot, E. Sontag, and T. Tuller. A model for competition for ribosomes in the cell. *Proc. Royal Society Interface*, 13:2015.1062, 2016.

[29] J. Miller, M. Al-Radhawi, and E. Sontag. Mediating ribosomal competition by splitting pools. *IEEE Control Systems Letters*, 5:1555–1560, 2020.

AD [*: CONTRIBUTED EQUALLY]: DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, NORTHEASTERN UNIVERSITY, BOSTON, MA, USA

Email address: atefedarabi93@gmail.com

ZA [*: CONTRIBUTED EQUALLY]: DIVISION OF GENETICS AND GENOMICS, DEPARTMENT OF PEDIATRICS, BOSTON CHILDREN'S HOSPITAL, BOSTON, MA, USA, HARVARD MEDICAL SCHOOL, BOSTON, MA, USA, AND BROAD INSTITUTE OF MIT AND HARVARD, CAMBRIDGE, MA, USA

Email address: z.an199401@gmail.com

MAR: DEPARTMENTS OF ELECTRICAL AND COMPUTER ENGINEERING AND BIOENGINEERING, NORTHEASTERN UNIVERSITY, BOSTON, MA, USA

Email address: malirdwi@gmail.com

WC: DEPARTMENT OF BIOENGINEERING, NORTHEASTERN UNIVERSITY, BOSTON, MA, USA

Email address: willwoncho@gmail.com

MS: DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, NORTHEASTERN UNIVERSITY, BOSTON, MA, USA

Email address: m.siami@northestern.edu

EDS: DEPARTMENTS OF ELECTRICAL AND COMPUTER ENGINEERING AND BIOENGINEERING, AND AFFILIATE OF DEPARTMENTS OF MATHEMATICS AND CHEMICAL ENGINEERING, NORTHEASTERN UNIVERSITY, BOSTON, MA, USA

Email address: sontag@sontaglab.org, e.sontag@northestern.edu, eduardo.sontag@gmail.com