Modular Machine Learning with Applications to Genetic Circuit Composition

Jichi Wang[1], Eduardo D. Sontag[2], and Domitilla Del Vecchio[3]

Abstract— In several applications, including in synthetic biology, one often has input/output data on a system composed of many modules, and although the modules' input/output functions and signals may be unknown, knowledge of the composition architecture can allow to significantly reduce the amount of training data required to learn the system's input/output mapping. Learning the modules' input/output functions is also necessary for designing new systems from different composition architectures. Here, we propose a modular learning framework, which incorporates prior knowledge of the system's compositional structure to (a) identify the composing modules' input/output functions from the system's input/output data and (b) achieve this by using a reduced amount of data compared to what would be required without knowledge of the compositional structure. To achieve this, we introduce the notion of *modular identifiability*, which allows to recover modules' input/output functions from a subset of system's input/output data, and provide theoretical guarantees on a class of systems motivated by genetic circuits. We demonstrate the theory on computational studies showing that a neural network (NNET) that accounts for the compositional structure can learn the composing modules' input/output functions and predict the system's output on inputs outside of the training set distribution. By reducing the need for experimental data and allowing modules' identification, this framework offers the potential to ease the design of synthetic biological circuits and of multi-module systems more generally.

I. Introduction

In synthetic biology, genetic circuits are commonly designed in a modular fashion, with each genetic module performing specific functions in isolation. However, when these modules are composed together in the cell, their performance can be significantly impacted by interactions with other modules, due to loading effects and resource competition between the modules [1]–[7]. These inter-dependencies complicate the characterization and prediction of a system's behavior. Researchers have been developing physics-based models to help design larger systems while accounting for or mitigating these context effects [3], [5], [8]–[12], as well as software tools [13], [14] to model genetic circuits with varying complexity of context descriptions. Yet, when composing genetic modules within the cellular host, there remain interactions that are difficult to model through first principles. Therefore, machine learning (ML) models, such as (recurrent) neural networks (NNETs) have been proposed to reduce the uncertainty of physics-based models of genetic circuits [15], [16], in-line with Physics-Informed Neural Networks (PINNs) approaches pioneered in [17].

These ML approaches applied to genetic circuit modeling, however, require large amounts of data. Also, they focus on identifying a mapping between the inputs and the outputs of a system composed of many genetic modules. Hence, they provide little information about the input/output mapping of the individual genetic modules. This information, in turn, would be useful for composing the modules in new arrangements to achieve new designs. Additionally, accounting for the composition architecture in an ML model may allow the use of substantially less training data to achieve the same predictive ability. Specifically, in the case of a system with multiple inputs, one may be able to train the ML model by activating one input at a time as opposed to requiring to generate data for all combinations of the inputs. If possible, this would simplify experiments and make data generation faster and cheaper.

In this paper, using information on the compositional structure, we propose a modular learning framework to identify the modules' input/output functions from training data, under certain assumptions on the model architecture. Specifically, we consider modules with unknown input/output function, whose outputs are then composed through a partially known map, called composition map, which has a known structure but unknown parameters. We investigate conditions under which one can learn the modules' input/output functions from data of the output of the composition map, called the global output, when the modules' inputs are only activated one at the time.

We demonstrate that, under *modular identifiability* conditions, this is possible. We thus show that using a NNET architecture that preserves the structure of the composition map, we can learn the modules' input/output functions and predict the global output for arbitrary combinations of inputs when the training set only considers one input being activated at the time. When using the same training data, a "monolithic" NNET that does not leverage the structure of the composition is unable to generalize the output prediction on arbitrary input combinations.

Related Work. The idea of training ML models on input/output data where inputs are turned on one at the time to then predict the output on a combination of inputs has been applied before in different synthetic biology settings,

[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA. Email: jichi@mit.edu

[2]Department of Electrical and Computer Engineering and Department of Bioengineering, Northeastern University, Boston, Massachusetts 02115, USA. Email: sontag@gmail.com

[3]Department of Mechanical Engineering and Biological Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA. Email: ddv@mit.edu

with promising results [18], [19]. However, these works did not provide conditions under which this is possible, which is one of the contributions of this paper.

In the context of networked dynamical systems, a related problem is to estimate local subsystem dynamics when the network topology is known. Under the assumptions that all the subsystems' output signals are linearly summed and some internal signals are directly measurable, the identifiability of the subsystems' LTI transfer functions have been addressed in [20], [21]. In the static case, closer to our problem formulation, identifiability conditions have been provided for subsystems' input/output nonlinear functions under the assumption that these are linearly composed [22], [23]. In our paper, instead, motivated by the composition architecture found in genetic circuits, we consider nonlinear composition of static nonlinear functions.

II. PROBLEM FORMULATION

A. General System Formulation

Consider a system Σ composed of n subsystems or modules. Each module is described by an unknown scalar input/output function $y_i = f_i(u_i)$, where $u_i \in \mathbb{R}$ denotes the input to the ith module. The global measured output $Y \in \mathbb{R}^n$ is given by a function $G \colon \mathbb{R}^n \to \mathbb{R}^m$, according to $Y = G(y_1, y_2, \ldots, y_n, \theta)$, in which $\theta \in \mathbb{R}^p$ is a parameter vector. We call this function the "composition map". The system structure is illustrated in Fig. 1.
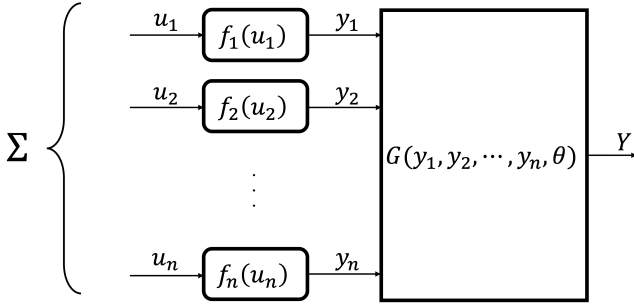

Fig. 1. Illustration of the system Σ. The system consists of n modules, each defined by $y_i = f_i(u_i)$ for $i \in \{1, \ldots, n\}$, whose outputs are fed into a composition map G.

In this paper, we will consider the problem of identifying the functions f_i, $i \in \{1, ..., n\}$, and parameter θ from measurements of $u = (u_1, \ldots, u_n) \in \mathbb{R}^n$ and Y. Specifically, we are interested in solving this problem when the inputs u_i are varied one at the time. To this end, let $a_i < b_i$ be real numbers, $U_i = [a_i, b_i] \subset \mathbb{R}$, and $u_i^* \in U_i$. Define the uni-modular input set $\mathcal{U} \subset \mathbb{R}^n$ as

$$\mathcal{U} := \bigcup_{i=1}^{n} \mathcal{U}_i, \tag{1}$$

where

$$\mathcal{U}_i := \left\{ u \in \mathbb{R}^n \,\middle|\, u_i \in U_i, u_j = u_j^*, \, \forall\, j \neq i \right\}. \tag{2}$$

Then, we ask when it is possible to identify the functions f_i for all $i \in \{1, ..., n\}$ from measurements of the output Y of system Σ for $u \in \mathcal{U}$.

Definition 1: The system Σ is said to be *modularly identifiable* on the uni-modular input set $\mathcal{U}$ if all functions $f_i(u_i)$ and parameters θ can be uniquely recovered from the measured output Y, that is:

$$G(\hat{f}_1(u_1), \ldots, \hat{f}_n(u_n), \hat{\theta}) = G(f_1(u_1), \ldots, f_n(u_n), \theta),$$
$$\forall\, u \in \mathcal{U}$$
$$\implies$$
$$\hat{f}_i(u_i) = f_i(u_i), \forall\, u_i \in U_i, i \in \{1, 2, \ldots, n\}, \text{ and } \hat{\theta} = \theta. \tag{3}$$

III. CASE OF A SINGLE MODULE

We first consider the case of a single module, i.e., $n = 1$, and one unknown parameter, $\theta \in \mathbb{R}$. Here, we consider the following model for the output Y:

$$Y = G(f(u), \theta) = \frac{\theta \cdot f(u)}{1 + f(u)}. \tag{4}$$
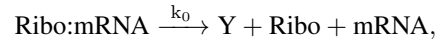
Without loss of generality, we let $a_i = 0$ and $b_i = 1$, such that $\mathcal{U} = U = [0, 1]$.

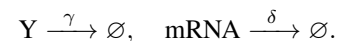### A. Motivation: Regulation of a Gene Expression Module

In this section, we illustrate how the form (4) emerges from the regulation of gene expression. The process of gene expression produces protein from DNA through two steps. In a first step, transcription transforms a DNA sequence into a messenger RNA (mRNA) sequence, and in a second step, translation transforms this mRNA sequence into a sequence of amino acids that then folds into a protein [24]. A protein, in turn, can be a transcriptional regulator, activating or repressing the transcription of other genes. These regulators, called transcription factors, can be regarded as the inputs to a gene expression module (Fig. 2). Each such module, can be described by a set of chemical reactions as follows (Chapter 2 of [24]). The process of transcription can be described by

$$\text{Transcription:} \quad \text{DNA} \xrightarrow{\bar{f}(u)} \text{mRNA} + \text{DNA},$$
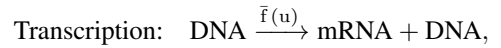
in which $\bar{f}(u)$ is a regulatory function called Hill function [24], which is increasing with u for an activator or decreasing with u for a repressor. Translation requires a cellular resource called the ribosome (Ribo), which is shared with the cellular translation processes, and can be written as:

$$\text{Translation:} \quad \text{mRNA} + \text{Ribo} \underset{\bar{d}}{\overset{\bar{a}}{\rightleftharpoons}} \text{Ribo:mRNA}$$

$$\text{Ribo:mRNA} \xrightarrow{k_0} Y + \text{Ribo} + \text{mRNA},$$

in which Y is the protein output of the gene expression module. Protein and mRNA then decay:

$$\text{Y} \xrightarrow{\gamma} \varnothing, \quad \text{mRNA} \xrightarrow{\delta} \varnothing.$$

We lump the rest of the cell mRNA into a single species $\text{mRNA}_{\text{cell}}$ which is produced from DNA_{cell} at a constant rate
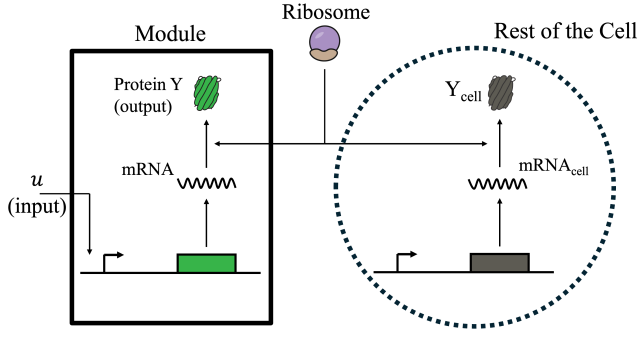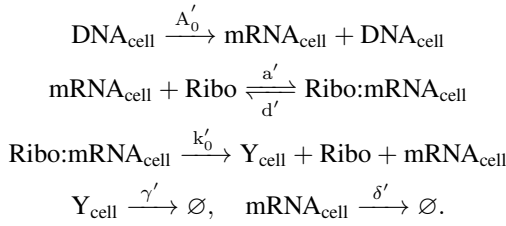
Fig. 2. Regulation of a single gene expression module.

A_0', and free ribosomes reversibly bind with $\text{mRNA}_{\text{cell}}$ to produce protein Y_{cell}, as done elsewhere [9], [24]:

$$\text{DNA}_{\text{cell}} \xrightarrow{A_0'} \text{mRNA}_{\text{cell}} + \text{DNA}_{\text{cell}}$$

$$\text{mRNA}_{\text{cell}} + \text{Ribo} \underset{d'}{\overset{a'}{\rightleftharpoons}} \text{Ribo:mRNA}_{\text{cell}}$$

$$\text{Ribo:mRNA}_{\text{cell}} \xrightarrow{k_0'} Y_{\text{cell}} + \text{Ribo} + \text{mRNA}_{\text{cell}}$$

$$Y_{\text{cell}} \xrightarrow{\gamma'} \varnothing, \quad \text{mRNA}_{\text{cell}} \xrightarrow{\delta'} \varnothing.$$

The Reaction Rate Equations (RREs) corresponding to these chemical reactions can be written using the law of mass action [24] as follows:

$$\frac{dY}{dt} = k_0 \cdot [\text{Ribo:mRNA}] - \gamma \cdot Y$$

$$\frac{d[\text{mRNA}]}{dt} = \bar{f}(u) \cdot [\text{DNA}]$$
$$- \bar{a} \cdot [\text{mRNA}] \cdot [\text{Ribo}] + \bar{d} \cdot [\text{Ribo:mRNA}]$$
$$+ k_0 \cdot [\text{Ribo:mRNA}] - \delta \cdot [\text{mRNA}]$$

$$\frac{d[\text{Ribo:mRNA}]}{dt} = \bar{a} \cdot [\text{mRNA}] \cdot [\text{Ribo}]$$
$$- \bar{d} \cdot [\text{Ribo:mRNA}] - k_0 \cdot [\text{Ribo:mRNA}].$$

Similarly, in the rest of the cell, we have the RREs as follow:

$$\frac{dY_{\text{cell}}}{dt} = k_0' \cdot [\text{Ribo:mRNA}_{\text{cell}}] - \gamma' \cdot Y_{\text{cell}}$$

$$\frac{d[\text{mRNA}_{\text{cell}}]}{dt} = A_0' \cdot [\text{DNA}_{\text{cell}}]$$
$$- a' \cdot [\text{mRNA}_{\text{cell}}] \cdot [\text{Ribo}] + d' \cdot [\text{Ribo:mRNA}_{\text{cell}}]$$
$$+ k_0' \cdot [\text{Ribo:mRNA}_{\text{cell}}] - \delta' \cdot [\text{mRNA}_{\text{cell}}]$$

$$\frac{d[\text{Ribo:mRNA}_{\text{cell}}]}{dt} = a' \cdot [\text{mRNA}_{\text{cell}}] \cdot [\text{Ribo}]$$
$$- d' \cdot [\text{Ribo:mRNA}_{\text{cell}}] - k_0' \cdot [\text{Ribo:mRNA}_{\text{cell}}].$$

Let R_T be the total concentration of ribosome, then we have the conservation law:

$$R_T = [\text{Ribo}] + [\text{Ribo:mRNA}] + [\text{Ribo:mRNA}_{\text{cell}}].$$

The binding and unbinding reactions are much faster than the catalytic reactions and gene expression, i.e., $R_T \cdot \bar{a}, \bar{d} \gg k_0, \gamma, \delta, \bar{f}(u)$ and $R_T \cdot a', d' \gg k_0', \gamma', \delta', A_0'$ [24]. Therefore, we can use the Quasi-Steady-State Approximation (QSSA) to set

$$\frac{d[\text{Ribo:mRNA}]}{dt} = 0 \text{ and } \frac{d[\text{Ribo:mRNA}_{\text{cell}}]}{dt} = 0.$$

Therefore, we have

$$[\text{Ribo:mRNA}] = \frac{[\text{mRNA}]}{K} \cdot [\text{Ribo}], \ K = \frac{\bar{d} + k_0}{\bar{a}},$$

$$[\text{Ribo:mRNA}_{\text{cell}}] = \frac{[\text{mRNA}_{\text{cell}}]}{K'} \cdot [\text{Ribo}], \ K' = \frac{d' + k_0'}{a'}.$$

Solving for the free ribosome using the conservation law for ribosome, we have that

$$[\text{Ribo}] = \frac{R_T}{1 + \frac{[\text{mRNA}_{\text{cell}}]}{K'} + \frac{[\text{mRNA}]}{K}}.$$

Then, we arrive at simplified differential equations for $\frac{dY}{dt}$, $\frac{d[\text{mRNA}]}{dt}$, and $\frac{d[\text{mRNA}_{\text{cell}}]}{dt}$:

$$\frac{dY}{dt} = k_0 \cdot \frac{[\text{mRNA}]}{K} \cdot \frac{R_T}{1 + \frac{[\text{mRNA}_{\text{cell}}]}{K'} + \frac{[\text{mRNA}]}{K}} - \gamma \cdot Y$$

$$\frac{d[\text{mRNA}]}{dt} = \bar{f}(u) \cdot [\text{DNA}] - \delta \cdot [\text{mRNA}]$$

$$\frac{d[\text{mRNA}_{\text{cell}}]}{dt} = A_0' \cdot [\text{DNA}_{\text{cell}}] - \delta' \cdot [\text{mRNA}_{\text{cell}}],$$

which is consistent with standard models [5]. At the unique equilibrium of the system, the Jacobian matrix is upper triangular with negative diagonal entries. Consequently, all eigenvalues are negative and the equilibrium is locally exponentially stable. Since we are interested in the steady state input/output mapping of the system, we then set the derivatives to zero to obtain the equilibrium and hence the global input/output function:

$$Y = \frac{\frac{k_0 \cdot [\text{DNA}] \cdot R_T}{\gamma \cdot \delta \cdot K} \cdot \bar{f}(u)}{1 + \frac{A_0' \cdot [\text{DNA}_{\text{cell}}]}{\delta' \cdot K'} + \frac{[\text{DNA}]}{\delta \cdot K} \cdot \bar{f}(u)}.$$

Dividing both the numerator and denominator by $(1 + \frac{A_0' \cdot [\text{DNA}_{\text{cell}}]}{\delta' \cdot K'})$, the protein expression level of a single transcriptional regulation module can be written as:

$$Y = \frac{\theta \cdot f(u)}{1 + f(u)},$$

in which $\theta = \frac{k_0 \cdot R_T}{\gamma}$ and $f(u) = \frac{[\text{DNA}]}{\delta \cdot K} / (1 + \frac{A_0' \cdot [\text{DNA}_{\text{cell}}]}{\delta' \cdot K'}) \cdot \bar{f}(u)$. This coincides with (4).

B. Modular Identifiability when $\theta = 1$

Here, we assume $\theta = 1$. Let $f, \hat{f} : C([0,1]) \to \mathbb{R}$, and $f(u), \hat{f}(u) \geq 0$ for $u \in [0,1]$, where $\hat{f}(u)$ is used to approximate $f(u)$. Let

$$G(f(u)) = \frac{f(u)}{1 + f(u)}.$$

Proposition 1: The model is *modularly identifiable*, that is:

$$G(f(u)) = G(\hat{f}(u)), \forall u \in [0,1]$$
$$\implies f(u) = \hat{f}(u), \forall u \in [0,1].$$

Proof: Since the function $G(s) = \frac{s}{1+s}$ is strictly increasing for $s \geq 0$, it is injective on its domain $\mathbb{R}^+$. By assumption, $f(u), \hat{f}(u)$ are positive for all $u \in [0,1]$. Thus, if $G(f(u)) = G(\hat{f}(u))$ for all $u \in [0,1]$, $f(u) = \hat{f}(u)$ for all $u \in [0,1]$. ∎

Proposition 2: For any $\epsilon > 0$, there exists a $\delta > 0$, such that

$$\left| G(f(u)) - G(\hat{f}(u)) \right| < \delta$$
$$\implies \left| f(u) - \hat{f}(u) \right| < \epsilon, \text{ for } u \in [0,1].$$

Proof: Since $G(s)$, $f(u)$, and $\hat{f}(u)$ are continuous, the sets $G(f([0,1]))$ and $G(\hat{f}([0,1]))$ are compact. Hence, their union $K := G(f([0,1])) \cup G(\hat{f}([0,1]))$ is a compact subset of $[0,1)$. Additionally, as $G(s)$ is injective, it has an inverse, given by

$$G^{-1}(y) = \frac{y}{1-y}, \ y \in K.$$

On the compact set K, the derivative

$$(G^{-1})'(y) = \frac{1}{(1-y)^2}$$

is continuous and thus bounded. This implies that G^{-1} is Lipschitz on K with Lipschitz constant $L > 0$. We thus have that

$$\left| f(u) - \hat{f}(u) \right| = \left| G^{-1}(G(f(u))) - G^{-1}(G(\hat{f}(u))) \right|$$
$$\leq L \left| G(f(u)) - G(\hat{f}(u)) \right|.$$

Thus, given any $\epsilon > 0$, by picking $\delta = \frac{\epsilon}{L}$, we prove the statement. ∎

Proposition 2 implies that if $G(f(u))$ can be estimated with sufficiently high accuracy, then $f(u)$ will likewise be estimated with comparable accuracy. We will illustrate this point in the next section with an example.

C. Example

Here, we assume to have input/output data (u, Y) for $Y = G(f(u))$ and $u \in [0,1]$ as training set, and use a ML model given by:

$$G(\hat{f}(u)) = \frac{\hat{f}(u)}{1 + \hat{f}(u)},$$

in which $\hat{f}(u)$ is a NNET. According to Proposition 2, if we make the error $|G(\hat{f}(u)) - G(f(u))|$ sufficiently small, we should also be making the error $|\hat{f}(u) - f(u)|$ sufficiently small. As an example, we let

$$f(u) = \frac{0.797 \cdot \left(\frac{u}{0.494}\right)^4}{1 + \left(\frac{u}{0.494}\right)^4} + 0.443.$$
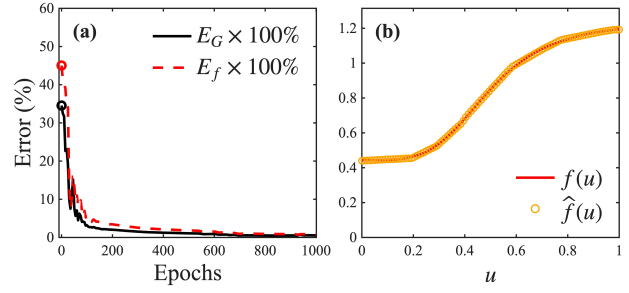

Fig. 3. Define $E_G := \max_k |G(\hat{f}(u_k)) - G(f(u_k))| / \max_k G(f(u_k))$ and $E_f := \max_k |\hat{f}(u_k) - f(u_k)| / \max_k f(u_k)$, in which u_k are the elements of the training set. (a) The plot shows the error convergence over 1000 epochs. The training dataset comprises 100 input/output pairs (u_k, Y_k), where inputs u_k is uniformly sampled from the interval $[0,1]$. We choose $\hat{f}(u)$ as a fully connected feedforward NNET with four hidden layers, each containing 20 ReLU-activated neurons, and initialize its weights using Kaiming initialization [25]. The NNET $\hat{f}(u)$ is trained using the Adam optimizer with a learning rate of 0.1 and full-batch gradient descent, with the cost function given by the mean square error $\frac{1}{100} \sum_{k=1}^{100} |G(\hat{f}(u_k)) - G(f(u_k))|^2$. (b) The plot shows the comparison between the trained $\hat{f}(u)$ and true $f(u)$ at the final (1000th) epoch.

As shown in Fig. 3(a), $|\hat{f}(u) - f(u)|$ converges to 0, as $|G(\hat{f}(u)) - G(f(u))|$ converges to 0, which verifies Proposition 2. Fig. 3(b) shows trained $\hat{f}(u)$ against the true $f(u)$ at the end epoch.

D. Modular Identifiability for Unknown θ

If θ is unknown, then we have an additional parameter to identify. In this case, a similar result as that in Section III-B can be obtained by restricting $f(u)$ and $\hat{f}(u)$ to be polynomials with given degree. Consequently, the system is modularly identifiable over the class of polynomial functions, which provides a sufficient condition for identifiability. When $f(u), \hat{f}(u)$ are, for instance, rational functions, we are not able to identify θ and $f(u)$. As a example, suppose that $\theta = 5$ and $f(u)$ is a Hill function:

$$f(u) = \frac{u}{1+u},$$

then any $\hat{\theta} \neq \theta$ and $\hat{f}(u) = \frac{\frac{\theta}{\hat{\theta}} \cdot f(u)}{1 + \frac{\hat{\theta}-\theta}{\theta} \cdot f(u)}$ will give $G(\hat{f}(u), \hat{\theta}) = G(f(u), \theta)$ but $\hat{f}(u) \neq f(u)$.

IV. CASE OF MULTIPLE MODULES

Without loss of generality, we consider two modules to simplify notation, that is, $n = 2$. The case of $n > 2$ modules can be treated similarly. Let

$$\mathcal{U}_1 = \{u \in \mathbb{R}^2 \mid u_1 \in [0,1], u_2 = 1\},$$
$$\mathcal{U}_2 = \{u \in \mathbb{R}^2 \mid u_1 = 1, u_2 \in [0,1]\},$$
$$\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2.$$

The measured output Y can be expressed as:

$$Y = G(f_1(u_1), f_2(u_2), \theta) = \begin{bmatrix} G_1(f_1(u_1), f_2(u_2), \theta) \\ G_2(f_1(u_1), f_2(u_2), \theta) \end{bmatrix}, \quad (5)$$
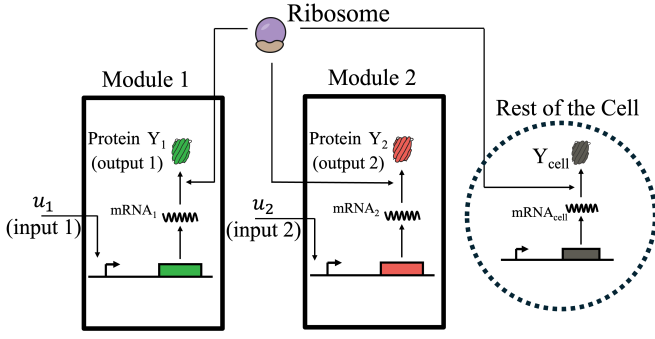
Fig. 4. Illustration showing two transcriptional regulation modules that share ribosomes.

with $\theta = (\theta_1, \theta_2) \in \mathbb{R}^2$ and

$$G_i(f_1(u_1), f_2(u_2), \theta) = \frac{\theta_i \cdot f_i(u_i)}{1 + f_1(u_1) + f_2(u_2)}, \quad i \in \{1, 2\}.$$
(6)

We make the following assumptions in the rest of the paper.

Assumption 1: Let $f_1, f_2 : [0, 1] \to \mathbb{R}$ be non-constant continuous functions such that $f_1(1) \neq 0$ and $f_2(1) \neq 0$. Furthermore, suppose $\theta \in \Theta$, where $\Theta \subset \mathbb{R}^2$ is compact.

A. Motivation: Regulation of Two Gene Expression Modules

In this case, we consider two transcriptional modules that are coupled due to competition for a shared cellular resource, the ribosome, illustrated in Fig. 4. The introduction of the second module gives rise to an additional regulatory function $\bar{f}_2(u_2)$. Similar derivations to those in Section III-A, in which we treat the mRNA of the second module analogously to the mRNA of the first module and use $\bar{f}_2(u_2)$ instead of $\bar{f}(u)$, yield Y in the form of (5)-(6).

B. Modular Identifiability

In this section, we give the main result of the paper, conditions that ensure we can estimate the functions $f_i(u_i)$, $i \in \{1, 2\}$, from measurements of Y on the input set $\mathcal{U}$ which varies the input for one module at the time.

Theorem 1: The model in (5)-(6) is *modularly identifiable*, that is:

$$G_i(\hat{f}_1(u_1), \hat{f}_2(u_2), \hat{\theta}) = G_i(f_1(u_1), f_2(u_2), \theta),$$
$$\forall u \in \mathcal{U}, \forall i \in \{1, 2\}$$
$$\implies$$

$$\hat{f}_i(u_i) = f_i(u_i) \text{ and } \hat{\theta}_i = \theta_i, \forall i \in \{1, 2\}, \forall u_i \in [0, 1].$$

Proof: Define the following constants:

$$A_1 = f_1(1), \hat{A}_1 = \hat{f}_1(1), A_2 = f_2(1), \hat{A}_2 = \hat{f}_2(1).$$

Also, define the following functions:
$G_1(f_1(u_1), f_2(u_2), \theta)$ for $(u_1, u_2) \in \mathcal{U}_1$ as:

$$G_{11}(u_1) = \frac{\theta_1 \cdot f_1(u_1)}{1 + f_1(u_1) + A_2}, u_1 \in [0, 1]$$

$G_2(f_1(u_1), f_2(u_2), \theta)$ for $(u_1, u_2) \in \mathcal{U}_1$ as:

$$G_{21}(u_1) = \frac{\theta_2 \cdot A_2}{1 + f_1(u_1) + A_2}, u_1 \in [0, 1]$$

$G_1(f_1(u_1), f_2(u_2), \theta)$ for $(u_1, u_2) \in \mathcal{U}_2$ as:

$$G_{12}(u_2) = \frac{\theta_1 \cdot A_1}{1 + A_1 + f_2(u_2)}, u_2 \in [0, 1]$$

$G_2(f_1(u_1), f_2(u_2), \theta)$ for $(u_1, u_2) \in \mathcal{U}_2$ as:

$$G_{22}(u_2) = \frac{\theta_2 \cdot f_2(u_2)}{1 + A_1 + f_2(u_2)}, u_2 \in [0, 1].$$

Let $\hat{G}_{ij}(u_j)$ for $i, j \in \{1, 2\}$ be defined as $G_{ij}(u_j)$ by replacing θ_i with $\hat{\theta}_i$, A_i with $\hat{A}_i$, and $f_j(u_j)$ with $\hat{f}_j(u_j)$. We then have

$$G_{11}(u_1) - \hat{G}_{11}(u_1) = 0 \implies$$
$$\theta_1 \cdot f_1(u_1) \cdot (1 + \hat{A}_2) - \hat{\theta}_1 \cdot \hat{f}_1(u_1) \cdot (1 + A_2) +$$
$$(\theta_1 - \hat{\theta}_1) \cdot f_1(u_1) \cdot \hat{f}_1(u_1) = 0, \quad (7)$$

and

$$G_{21}(u_1) - \hat{G}_{21}(u_1) = 0 \implies$$
$$\theta_2 \cdot A_2 \cdot (1 + \hat{f}_1(u_1)) - \hat{\theta}_2 \cdot \hat{A}_2 \cdot (1 + f_1(u_1)) +$$
$$(\theta_2 - \hat{\theta}_2) \cdot A_2 \cdot \hat{A}_2 = 0. \quad (8)$$

From (7) and (8), we can obtain two expressions for $\hat{f}(u_1)$. Then, by equating these two expressions and rearranging the terms, we obtain

$$\alpha \cdot f_1(u_1)^2 + \beta \cdot f_1(u_1) + \gamma = 0, \quad (9)$$

where

$\alpha = \frac{\hat{\theta}_2 \hat{A}_2}{\theta_2 A_2}(\theta_1 - \hat{\theta}_1),$

$\beta = (\theta_1 - \hat{\theta}_1)\frac{\hat{\theta}_2 \hat{A}_2}{\theta_2 A_2} + \theta_1(1 + \hat{A}_2)$
$\quad - \hat{\theta}_1(1 + A_2)\frac{\hat{\theta}_2 \hat{A}_2}{\theta_2 A_2} - (\theta_1 - \hat{\theta}_1)\left(\frac{\theta_2 - \hat{\theta}_2}{\theta_2}\hat{A}_2 + 1\right),$

$\gamma = -\hat{\theta}_1(1 + A_2)\frac{\hat{\theta}_2 \hat{A}_2}{\theta_2 A_2} + \hat{\theta}_1(1 + A_2)\left(\frac{\theta_2 - \hat{\theta}_2}{\theta_2}\hat{A}_2 + 1\right).$

Since $f_1(u_1)$ is continuous and not a constant, its image $f_1([0, 1])$, by the Intermediate Value Theorem, is an interval in $\mathbb{R}$. Therefore, the left-hand side of (9) is a polynomial in $f_1(u_1)$ that vanishes on a subset of $\mathbb{R}$ that has infinitely many limit points. Then, using the Polynomial Identity Theorem [26], we have that all the coefficients of $f_1(u_1)^2$, $f_1(u_1)$, and $f_1(u_1)^0$ must be 0. Then, setting $\alpha = 0$ gives $\hat{\theta}_1 = \theta_1$. Using the same argument for $G_{12}(u_2)$ and $G_{22}(u_2)$, we can obtain that $\hat{\theta}_2 = \theta_2$. Setting $\beta = 0$ and $\gamma = 0$ with $\hat{\theta}_1 = \theta_1$ and $\hat{\theta}_2 = \theta_2$, we then have

$$(1 + \hat{A}_2) - (1 + A_2) \cdot \frac{\hat{A}_2}{A_2} = 0 \implies \hat{A}_2 = A_2.$$

Thus, it follows from (8) that $\hat{f}_1(u_1) = f_1(u_1)$ for all $u_1 \in [0, 1]$. Again, applying this argument to $G_{12}(u_2)$ and $G_{22}(u_2)$, we obtain $\hat{f}_2(u_2) = f_2(u_2)$ for all $u_2 \in [0, 1]$. ∎

In the next theorem, we show that if the approximating functions $G_1(\hat{f}_1(u_1), \hat{f}_2(u_2), \hat{\theta})$ and $G_2(\hat{f}_1(u_1), \hat{f}_2(u_2), \hat{\theta})$ are arbitrarily close to the true functions on the input set $\mathcal{U}$, then $\hat{f}_1(u_1), \hat{f}_2(u_2)$ will be close to $f_1(u_1), f_2(u_2)$.

Theorem 2: For any $\epsilon > 0$, there exists a $\delta > 0$, such that if

$$\left| G_i(\hat{f}_1(u_1), \hat{f}_2(u_2), \hat{\theta}) - G_i(f_1(u_1), f_2(u_2), \theta) \right| < \delta,$$
$$\forall\, i \in \{1, 2\}, \forall\, u \in \mathcal{U},$$

then

$$\left| \hat{f}_i(u_i) - f_i(u_i) \right| < \epsilon \text{ and } \left| \hat{\theta}_i - \theta_i \right| < \epsilon,$$
$$\forall\, i \in \{1, 2\}, \forall\, u_i \in U_i.$$

Proof: Let $(u_i^1, u_i^2) \in [0, 1]^2$ for $i \in \{1, 2\}$. Let $x = (f_1(u_1^1), f_1(u_1^2), f_1(1), f_2(u_2^1), f_2(u_2^2), f_2(1), \theta_1, \theta_2) \in \mathbb{R}^8$. When u_i^1, u_i^2 range in $[0, 1]$ and θ_1, θ_2 range in Θ, from Assumption 1, $x \in \mathcal{X}$ with $\mathcal{X}$ a compact subset of $\mathbb{R}^8$. Define the map $\mathcal{F} : \mathcal{X} \to \mathbb{R}^8$ by $\mathcal{F}(x) = y$, where

$$y = \begin{bmatrix} \dfrac{\theta_1 f_1(u_1^1)}{1 + f_1(u_1^1) + f_2(1)} \\ \dfrac{\theta_2 f_2(1)}{1 + f_1(u_1^1) + f_2(1)} \\ \dfrac{\theta_1 f_1(1)}{1 + f_1(1) + f_2(u_2^1)} \\ \dfrac{\theta_2 f_2(u_2^1)}{1 + f_1(1) + f_2(u_2^1)} \\ \dfrac{\theta_1 f_1(u_1^2)}{1 + f_1(u_1^2) + f_2(1)} \\ \dfrac{\theta_2 f_2(1)}{1 + f_1(u_1^2) + f_2(1)} \\ \dfrac{\theta_1 f_1(1)}{1 + f_1(1) + f_2(u_2^2)} \\ \dfrac{\theta_2 f_2(u_2^2)}{1 + f_1(1) + f_2(u_2^2)} \end{bmatrix}.$$

Let $\mathcal{Y} = \mathcal{F}(\mathcal{X})$ be the image of $\mathcal{X}$. Since $\mathcal{F}$ is continuous and $\mathcal{X}$ is compact, $\mathcal{Y}$ is compact. It can be shown that $\mathcal{F}$ is injective (see Lemma 1 in Appendix), and hence $\mathcal{F}: \mathcal{X} \to \mathcal{Y}$ is a bijection. Thus, by Theorem 26.6 in [27], $\mathcal{F}: \mathcal{X} \to \mathcal{Y}$ is a homeomorphism, that is, $\mathcal{F}^{-1}$ is continuous on $\mathcal{Y}$. Since $\mathcal{Y}$ is a compact set, $\mathcal{F}^{-1}: \mathcal{Y} \to \mathcal{X}$ is uniformly continuous on $\mathcal{Y}$ (Theorem 27.6 in [27]). This implies that for all $\epsilon > 0$, there exists $\delta > 0$, independent of $y_1, y_2 \in \mathcal{Y}$, such that

$$\| y_1 - y_2 \| < \delta \implies \| x_1 - x_2 \| < \epsilon,$$

with $x_1 = \mathcal{F}^{-1}(y_1)$ and $x_2 = \mathcal{F}^{-1}(y_2)$.

Hence, for all $\epsilon > 0$, there exists $\delta > 0$, such that if $|G_i(\hat{f}_1(u_1), \hat{f}_2(u_2), \hat{\theta}) - G_i(f_1(u_1), f_2(u_2), \theta)| < \delta$ for all $u \in \mathcal{U}$ and $i \in \{1, 2\}$, then for all $i \in \{1, 2\}$ and for all $u_i \in [0, 1]$, we have that $|\hat{f}_i(u_i) - f_i(u_i)| < \epsilon$ and $|\hat{\theta}_i - \theta_i| < \epsilon$. ∎

C. Computational Study

Here, we demonstrate that, by virtue of Theorem 1 and 2, we can learn the subsystems' functions f_1, f_2 and the parameters θ_1, θ_2 from measurements of the global output Y on the input set $\mathcal{U}$. In particular, we define the following

functions and parameters. Let $f_1(u_1)$ be an activating Hill function and $f_2(u_2)$ be a repressing Hill function.

$$f_1(u_1) = \frac{0.326 \cdot \left(\frac{u_1}{0.952} \right)^4}{1 + \left(\frac{u_1}{0.952} \right)^4} + 0.176,$$

$$f_2(u_2) = \frac{0.261}{1 + \left(\frac{u_2}{0.415} \right)^2} + 0.192,$$

$$\theta_1 = 0.703, \qquad \theta_2 = 0.204.$$

As our ML model, we take $\hat{f}_1, \hat{f}_2$ as NNETs and $\hat{\theta}_1, \hat{\theta}_2$ as learnable parameters. Our training dataset consists of 200 pairs (u_k, Y_k) with $u_k \in \mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$. The first 100 inputs are of the form $(\bar{u}_1, 1)$, where $\bar{u}_1$ is uniformly sampled from $[0, 1]$. Similarly, the second 100 inputs are of the form $(1, \bar{u}_2)$, where $\bar{u}_2$ is uniformly sampled from $[0, 1]$. We optimize the following cost function:

$$\frac{1}{200} \sum_{k=1}^{200} \left| G_1(\hat{f}(u_{1_k}), \hat{f}(u_{2_k}), \hat{\theta}) - G_1(f(u_{1_k}), f(u_{2_k}), \theta) \right|^2$$
$$+ \left| G_2(\hat{f}(u_{1_k}), \hat{f}(u_{2_k}), \hat{\theta}) - G_2(f(u_{1_k}), f(u_{2_k}), \theta) \right|^2. \tag{10}$$

Here, $\hat{f}_1(u_1), \hat{f}_2(u_2)$ are chosen as fully connected feedforward NNETs with four hidden layers, each containing 20 ReLU-activated neurons. The weights of $\hat{f}_1(u_1), \hat{f}_2(u_2)$ are initialized using Kaiming initialization [25], and $\hat{\theta}_1, \hat{\theta}_2$ are initialized as 3. Training is performed using the Adam optimizer with a learning rate of 0.005 and full-batch gradient descent.

As shown in Fig. 5(a)-(c), for $i \in \{1, 2\}$, $|\hat{f}_i(u_i) - f_i(u_i)|$ and $|\hat{\theta}_i - \theta_i|$ converge to 0, as $|G_i(\hat{f}_1(u_1), \hat{f}_2(u_2), \hat{\theta}) - G_i(f_1(u_1), f_2(u_2), \theta)|$, $i \in \{1, 2\}$, converge to 0, which verifies Theorem 2. The comparison between the learned model $\hat{f}_1(u_1), \hat{f}_2(u_2)$ and the true functions $f_1(u_1), f_2(u_2)$ at the final epoch is shown in Fig. 5(d).

Prediction of system's output on out-of-distribution test data. Because the estimated modules' functions and parameter θ are very close to the true entities, we expect that our learned model, trained on the input set $\mathcal{U}$ in which one input only is varied at the time, should predict well the output from arbitrary input combinations. These input combinations form test data that lie outside the distribution of the training data. We therefore evaluate the learned model prediction on this out-of-distribution test data against the predictions of a monolithic NNET trained on the same input set $\mathcal{U}$. This is shown in Fig 6. The monolithic ML model is taken to be a fully connected feedforward NNET with two inputs, two outputs, and four hidden layers of 50 ReLU-activated neurons each, with the weights initialized using Kaiming initialization. We used the Adam optimizer with a learning rate of 0.001 to minimize the same error in (10), training the model for 8,000 epochs until the training error converged.

From these plots, we see that while the modular learning model generalizes well on the entire input set, the monolithic approach does not generalize well for arbitrary combinations
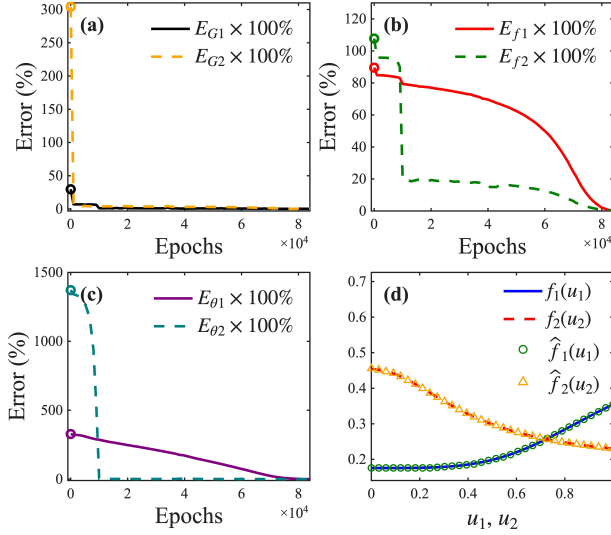
Fig. 5. Error convergence over 84,000 epochs for the case of two modules. Define the following errors: $E_{Gi} := \max_k |G_i(\hat{f}_1(u_{1_k}), \hat{f}_2(u_{2_k}), \hat{\theta}) - G_i(f_1(u_{1_k}), f_2(u_{2_k}), \theta)| / \max_k G_i(f_1(u_{1_k}), f_2(u_{2_k}), \theta)$, $E_{fi} := \max_k |\hat{f}_i(u_{i_k}) - f_i(u_{i_k})| / \max_k f_i(u_{i_k})$, and $E_{\theta i} := |\hat{\theta}_i - \theta_i| / \theta_i$, for $i = \{1, 2\}$. (a) The plot shows the convergence of E_{G1} and E_{G2}. (b) The plot shows the convergence of E_{f1} and E_{f2}. (c) The plot shows the convergence of $E_{\theta 1}$ and $E_{\theta 2}$. (d) The plot shows the comparison between the true functions $f_1(u_1)$, $f_2(u_2)$ and their learned counterparts $\hat{f}_1(u_1)$, $\hat{f}_2(u_2)$ at the final (84,000th) epoch. Also, at the final epoch, the learned parameters $\hat{\theta}_1 = 0.70325, \hat{\theta}_2 = 0.20361$ vs. the true parameters $\theta_1 = 0.70340, \theta_2 = 0.20357$.
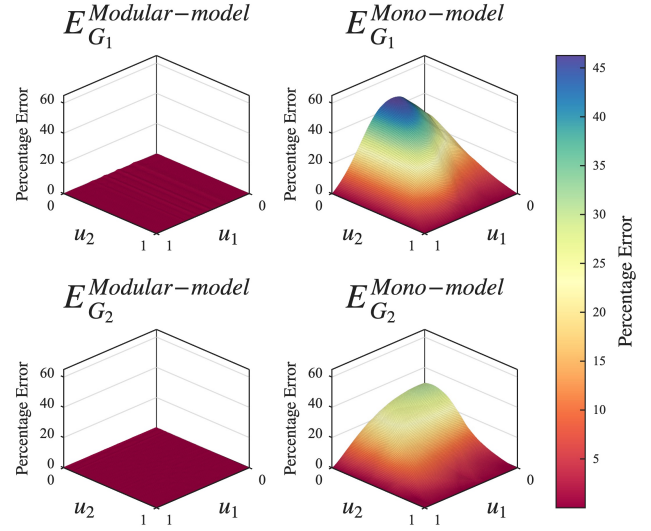

Fig. 6. We use the same simulation setup as in Fig 5 to compare the generalization ability of our modular learning approach and a monolithic learning approach. The modular learning model is the same as in Fig 5. Let *Modular-model* denote the trained modular learning model and *Mono-model* be the trained monolithic learning model. For $M \in \{\textit{Modular-model}, \textit{Mono-model}\}$ and $i \in \{1, 2\}$, the point-wise error is defined as $E_{G_i}^M(u_1, u_2) := |M(u_1, u_2)_i - G_i(f_1(u_1), f_2(u_2), \theta)| / G_i(f_1(u_1), f_2(u_2), \theta)$. The loss surfaces are generated from 10,000 test grid points uniformly sampled over $[0, 1]^2$.

of inputs, but only for those input combinations that belong to the distribution of the training set.

V. CONCLUSION

In this paper, we introduced a modular ML framework, which leverages prior knowledge of the composition structure of a system to learn the input/output functions of the composing modules from input/output data of the system. We have also demonstrated how, by learning the modules, the modular ML model can generalize on out-of-distribution data, whereas a monolithic ML model fails to do so.

As a running example, we considered a system architecture that emerges when composing genetic circuit modules in the cell, which share resources. In this setup, we have considered an example involving several input/output genetic modules operating in the cell and competing for ribosomes. We have demonstrated that it is sufficient to activate one module at the time to enable predictions on arbitrary combinations of modules' input. From a practical point of view, this allows to dramatically reduce the data requirement for enabling prediction of the behavior of many modules operating together in the cell while accounting for the effects of context-dependence [1]–[7]. In future work, we seek to extend these ML models to capture more context effects beyond resource sharing and to use these for system design.

Our work, at present, idealizes perfect measurement of global input/output data. In real experimental settings, however, measurements can be noisy, which may cause the

learned functions and parameters to deviate from their true values. As a direction for future work, it would be interesting to investigate how the modular learning approach performs when applied to such real-world data. In addition, while this paper focuses on the steady-state input/output mapping, extending the framework to a dynamical systems setting could provide deeper insights into the system's behavior over time-series input/output data.

APPENDIX

Lemma 1: Assume $f_1(u_1^1) \neq f_1(u_1^2)$, $f_2(u_2^1) \neq f_2(u_2^2)$ and $\theta_1, \theta_2 \neq 0$. Define, for $j, k \in \{1, 2\}$,

$$G_{11}^j := \frac{\theta_1 f_1(u_1^j)}{1 + f_1(u_1^j) + f_2(1)}, \quad G_{21}^j := \frac{\theta_2 f_2(1)}{1 + f_1(u_1^j) + f_2(1)},$$

$$G_{12}^k := \frac{\theta_1 f_1(1)}{1 + f_1(1) + f_2(u_2^k)}, \quad G_{22}^k := \frac{\theta_2 f_2(u_2^k)}{1 + f_1(1) + f_2(u_2^k)}.$$

Consider the map

$$\mathcal{F} : \big(f_1(u_1^1), f_1(u_1^2), f_1(1), f_2(u_2^1), f_2(u_2^2), f_2(1), \ \theta_1, \theta_2\big) \\ \to \big(G_{11}^1, G_{21}^1, G_{12}^1, G_{22}^1, G_{11}^2, G_{21}^2, G_{12}^2, G_{22}^2\big).$$

Then, $\mathcal{F}$ is injective.

Proof: For $j, k \in \{1, 2\}$ we obtain the ratios

$$\frac{G_{11}^j}{G_{21}^j} = \frac{\theta_1}{\theta_2} \frac{f_1(u_1^j)}{f_2(1)}, \quad \frac{G_{22}^k}{G_{12}^k} = \frac{\theta_2}{\theta_1} \frac{f_2(u_2^k)}{f_1(1)}. \tag{11}$$

We also compute

$$\frac{\theta_2}{G_{21}^j} = \frac{1}{f_2(1)} + 1 + \frac{f_1(u_1^j)}{f_2(1)} = \frac{1}{f_2(1)} + 1 + \frac{\theta_2}{\theta_1}\frac{G_{11}^j}{G_{21}^j}, \quad (12)$$

and

$$\frac{\theta_1}{G_{12}^k} = \frac{1}{f_1(1)} + 1 + \frac{f_2(u_2^k)}{f_1(1)} = \frac{1}{f_1(1)} + 1 + \frac{\theta_1}{\theta_2}\frac{G_{22}^k}{G_{12}^k}. \quad (13)$$

First, we recover θ_1. Subtracting (12) for $j = 1$ and $j = 2$ and dividing by θ_2 on both sides, we get

$$\theta_1 = \frac{\frac{G_{11}^1}{G_{21}^1} - \frac{G_{11}^2}{G_{21}^2}}{\frac{1}{G_{21}^1} - \frac{1}{G_{21}^2}}.$$

Similarly, we obtain θ_2 through (13). Then, with θ_1, θ_2 both known, we can use (12) and (13) to solve for $f_2(1), f_1(1)$. Finally, from the ratios in (11) we can derive $f_1(u_1^1), f_1(u_1^2), f_2(u_2^1), f_2(u_2^2)$. ∎

REFERENCES

[1] S. Jayanthi, K. S. Nilgiriwala, and D. D. Vecchio, "Retroactivity controls the temporal dynamics of gene transcription," *ACS Synthetic Biology*, vol. 2, no. 8, pp. 431–441, 2013.

[2] T. W. Grunberg and D. D. Vecchio, "Modular analysis and design of biological circuits," *Current Opinion in Biotechnology*, vol. 63, pp. 41–47, 2020.

[3] D. Mishra, P. M. Rivera, A. Lin, D. D. Vecchio, and R. Weiss, "A load driver device for engineering modularity in biological networks," *Nature Biotechnology*, vol. 32, no. 12, pp. 1268–1275, 2014.

[4] D. D. Vecchio, A. J. Ninfa, and E. D. Sontag, "Modular cell biology: retroactivity and insulation," *Molecular Systems Biology*, vol. 4, no. 1, p. 161, 2008.

[5] Y. Qian, H.-H. Huang, J. I. Jiménez, and D. D. Vecchio, "Resource competition shapes the response of genetic circuits," *ACS Synthetic Biology*, vol. 6, no. 7, pp. 1263–1272, 2017.

[6] R. D. Blasi, J. Gabrielli, K. Shabestary, I. Ziarti, *et al.*, "Understanding resource competition to achieve predictable synthetic gene expression in eukaryotes," *Nature Reviews Bioengineering*, vol. 2, no. 9, pp. 721–732, 2024.

[7] R. D. Blasi, M. Pisani, F. Tedeschi, M. M. Marbiah, K. Polizzi, S. Furini, V. Siciliano, and F. Ceroni, "Resource-aware construct design in mammalian cells," *Nature Communications*, vol. 14, no. 1, p. 3576, 2023.

[8] D. D. Vecchio, "Modularity, context-dependence, and insulation in engineered biological circuits," *Trends in Biotechnology*, vol. 33, no. 2, pp. 111–119, Feb 2015.

[9] A. György, J. I. Jiménez, J. Yazbek, H.-H. Huang, H. Chung, R. Weiss, and D. D. Vecchio, "Isocost lines describe the cellular economy of genetic circuits," *Biophysical Journal*, vol. 109, no. 3, pp. 639–646, 2015.

[10] H.-H. Huang, Y. Qian, and D. D. Vecchio, "A quasi-integral controller for adaptation of genetic modules to variable ribosome demand," *Nature Communications*, vol. 9, p. 5415, 2018.

[11] R. D. Jones, Y. Qian, V. Siciliano, B. DiAndreth, J. Huh, R. Weiss, and D. D. Vecchio, "An endoribonuclease-based feedforward controller for decoupling resource-limited genetic modules in mammalian cells," *Nature Communications*, vol. 11, p. 5690, 2020.

[12] T. Frei, F. Cella, F. Tedeschi, J. Gutiérrez, G.-B. Stan, M. Khammash, and V. Siciliano, "Characterization and mitigation of gene expression burden in mammalian cells," *Nature Communications*, vol. 11, p. 4641, 2020.

[13] A. Pandey, W. Poole, A. Swaminathan, V. Hsiao, and R. M. Murray, "Fast and flexible simulation and parameter estimation for synthetic biology using bioscrape," *Journal of Open Source Software*, vol. 8, no. 83, p. 5057, 2023.

[14] W. Poole, A. Pandey, A. Shur, Z. A. Tuza, and R. M. Murray, "Biocrnpyler: Compiling chemical reaction networks from biomolecular parts in diverse contexts," *PLoS Computational Biology*, vol. 18, no. 4, p. e1009987, 2022.

[15] A. Darabi, Z. An, M. A. Al-Radhawi, W. Cho, M. Siami, and E. D. Sontag, "Combining model-based and data-driven models: an application to synthetic biology resource competition," *bioRxiv*, 2025, preprint, posted March 2025.

[16] S. Palacios, J. J. Collins, and D. Del Vecchio, "Machine learning for synthetic gene circuit engineering," *Current Opinion in Biotechnology*, vol. 92, p. 103263, 2025.

[17] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[18] M. Eslami, A. E. Borujeni, H. Eramian, M. Weston, G. Zheng, J. Urrutia, C. Corbet, D. Becker, K. Maschhoff, A. Clowers, A. Cristofaro, H. D. Hosseini, D. B. Gordon, Y. Dorfan, J. Singer, M. Vaughn, N. Gaffney, J. Fonner, C. A. V. Stubbs, and E. Yeung, "Prediction of whole-cell transcriptional response with machine learning," *Bioinformatics*, vol. 38, no. 2, pp. 404–409, 2022.

[19] M. A. Alcantar, M. A. English, J. A. Valeri, and J. J. Collins, "A high-throughput synthetic biology approach for studying combinatorial chromatin-based transcriptional regulation," *Molecular Cell*, vol. 84, no. 12, pp. 2382–2396.e9, 2024.

[20] P. M. J. van den Hof, A. G. Dankers, P. S. C. Heuberger, and X. J. A. Bombois, "Identification of dynamic models in complex networks with prediction error methods: Basic methods for consistent module estimates," *Automatica*, vol. 49, no. 10, pp. 2994–3006, 2013.

[21] A. Dankers, P. M. J. V. den Hof, X. Bombois, and P. S. C. Heuberger, "Identification of dynamic models in complex networks with prediction error methods: Predictor input selection," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 937–952, 2016.

[22] R. Vizuete and J. M. Hendrickx, "Nonlinear network identifiability: The static case," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 443–448.

[23] ——, "Nonlinear network identifiability with full excitations," *arXiv preprint arXiv:2405.07636*, 2024.

[24] D. Del Vecchio and R. M. Murray, *Biomolecular Feedback Systems*. Princeton, NJ, USA: Princeton University Press, 2014.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[26] T. W. Hungerford, *Algebra*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1974, vol. 73, volume 73 in GTM series.

[27] J. R. Munkres, *Topology*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2000.