# Learning-based Reconstruction of Chemical Reaction Networks

**Juechun Tang**
Chemical and Biological Engineering
Princeton University
juechunt@princeton.edu

**Hongtao Zhong**
Mechanical and Aerospace Engineering
Princeton University
hongtaoz@princeton.edu

## Abstract

Dynamics of chemical reaction networks (CRNs) are of central importance for understanding various phenomena including explosion, catalysis, and biological systems. Traditionally, the determination of rate constants of individual reactions in CRNs requires much prior knowledge, complicated numerical integration, and contains large uncertainty. In this work, we propose to apply machine learning techniques including linear regression, recurrent neural network (RNN), and Gaussian process (GP) model to determine the reaction rate constants and reconstruct chemical reaction networks from discrete experimental data. We showed that RNN best reserved the system dynamics from small to large-scale systems, while linear regression with poly-terms performed best in simple kinetics but failed to capture nonlinear dynamics in real systems. With the promising performance of RNN in fully-observed systems, our work can be further applied and tested in partially-observed nonlinear systems, potentially facilitating research in complex chemical kinetics.

## 1 Introduction

Reaction systems with complex chemical kinetics give rise to complicated yet important phenomena across science and engineering. Taking the combustion process for example, some combustion systems may induce violent uncontrollable phenomena like explosion or wild fires, while others may lead to controllable oxidation in applications like industrial burners, engines, and power plants. Therefore, it's of great importance to systematically study such reaction systems and their related dynamic properties.

Chemical reaction network (CRN) is one approach for reaction systems. Since its foundation in the 1960s [1], it has attracted a growing research interest, mainly due to its broad applications in chemical engineering and theoretical chemistry. A CRN is defined as a set of chemical species $\mathbf{S}$ and a set of reactions $\mathbf{R}$. Species concentrations $\mathbf{Y}$ in the such systems are evolved as

$$\frac{\mathrm{d}\mathbf{Y_i}}{\mathrm{dt}} = \sum_{\mathrm{j}} \mathbf{w_{i,j}} = \mathbf{f}(\mathbf{Y_i}, \mathbf{k_j}) \qquad \mathrm{i} \in \mathbf{S}, \quad \mathrm{j} \in \mathbf{R} \tag{1}$$

where $\mathrm{w_{i,j}}$ denotes the reaction rate of species i contributed from the reaction j. Individual $\mathrm{w_{i,j}}$ is modeled by the law of mass action as $\mathrm{w_{i,j}} = \mathrm{a_{i,j}} \mathrm{k_j} \prod_{\mathrm{i}^*} \mathrm{Y_i}$, where $\mathrm{a_{i,j}}$ is the stoichiometric coefficient, $\mathrm{k_j}$ is the reaction rate constant for the reaction j and $\mathrm{i}^*$ denotes all the reactants involved in the reaction j.

From the above formulation, CRNs are governed by systems of ordinary differential equation (ODEs). Key parameters are the reaction rate constants $\mathrm{k_j}$, which control the nonlinear interaction between species. Traditionally, the determination of rate constants $\mathrm{k_j}$ is achieved by trail-and-error fitting process. Time-consuming trial runs for different $\mathrm{k_j}$ and complicated numerical integration are

required for this method, which motivate us seeking for more efficient and automatic algorithms to determine the reaction rate constant $k_j$. Especially since chemical kinetics are usually complex and nonlinear in nature, an autonomous learning process would greatly facilitate research in complex reaction networks.

In this work, we proposed to apply machine learning techniques including linear regression, Gaussian process (GP), and recurrent neural network (RNN) to automatically determine reaction rate constants for CRNs, i.e., infer the parameters for governing ODE systems using a series of observations of $Y_i$ at different time points. We analyzed three CRN systems, the atmospheric $O_2$-$O_3$-O kinetics, biochemical kinetic oscillation, and the reduced $H_2-O_2$ combustion mechanism with learning-based methods and discussed possible future works.

## 2 Related Work

### 2.1 CRN Examples from Real World

CRNs are typically complex chemical systems, playing an active part in all branches of chemistry. Here we introduce some background for the three CRN systems we analyze in this work.

In atmospheric chemistry, $O_2$ - $O_3$ - O kinetics is an indispensable component for ozone destruction and global climate change [2]. $O_3$ is continually regenerated in Earth's atmosphere, mostly in the tropical upper stratosphere and mesosphere. The reduced mechanism is written as

$$\text{O}_3 \text{ Creation} \quad \text{O}_2 \xrightarrow{h\nu} \text{O}\cdot + \text{O}\cdot \quad \text{(R1)} \qquad \text{O}\cdot + \text{O}_2 \longrightarrow \text{O}_3 \quad \text{(R2)}$$

$$\text{O}_3 \text{ Removal} \quad \text{O}_3 \xrightarrow{h\nu} \text{O}_2 + \text{O}\cdot \quad \text{(R3)} \qquad \text{O}_3 + \text{O}\cdot \longrightarrow \text{O}_2 + \text{O}_2 \quad \text{(R4)}$$

The production of ozone and free radicals O are highly dependent of ultraviolet radiation (UV) from the sun. Therefore, we can evaluate the UV radiation intensity and ozone depletion by the reaction rate constants for R1-R4. This reduced system includes 3 species and 4 reactions. The simple structure makes it suitable as an starting case to validate our learning models.

The second CRN example is from biological chemistry, showing the circadian oscillation by KaiC phosphorylation. This cyanobacterial circadian clock is an oscillating biochemical system that involves only 3 core clock proteins: KaiA, KaiB, and KaiC. When supplemented with ATP, the system generates a robust 22hr-period oscillation. Briefly, KaiC autophosphorylates on 2 residues, S431 or/and T432, producing 3 phospho-forms of KaiC: T, D (represented as ST in the figure), and S (Fig.1). Rust et al [3] showed that the phosphorylation is greatly increased with KaiA in the system and the addition of KaiB, which inhibits kaiA:KaiC phosphorylation, induces the oscillatory behavior. We constructed the ODE model as shown below (parameters obtained from [3]):

$$\frac{dT}{dt} = K_{UT}U + K_{DT}D - K_{TU}T - K_{TD}T$$
$$\frac{dD}{dt} = K_{TD}T + K_{SD}S - K_{DT}D - K_{DS}D$$
$$\frac{dS}{dt} = K_{US}U + K_{DS}D - K_{SU}S - K_{SD}S$$
$$A = \max\{0, [KaiA] - 2mS\}$$
$$K_{XY} = k^0_{XY} + \frac{k^A_{XY}A}{K_{1/2} + A}$$

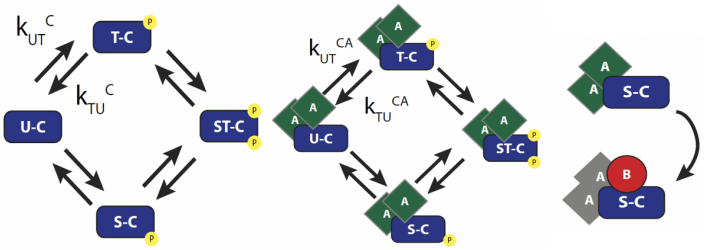



Figure 1: KaiC Phosphorylation Kinetics Model and Governing Equations

The rate constant for the transition from state X to state Y is denoted $k_{XY}$ , with $k^0_{XY}$ being the rate in the absence of KaiA and $k^A_{XY}$ being the influence of KaiA.

The $H_2-O_2$ combustion mechanism is the last CRN example demonstrated in this work. $H_2$ is the simplest but also a major fuel used in combustion. Hydrogen oxidation mechanism is crucial to understand because it always shows up as a sub-component in any of a larger systems of hydrocarbons, making rate constants for $H_2-O_2$ combustion important and fundamental for understanding

any hydrocarbon combustion. The list of reactions are shown in Table 1. $H_2-O_2$ combustion is activated at high temperature. The rate constants has strong nonlinear relations with pressure and temperature. For simplicity, we assume that reaction rate constants for all reactions remain constant in a constant-temperature, constant-pressure reaction system (T=1000 K, P = 1 atm).

### 2.2 Current Methodology

Currently, the determination of rate constants mainly depends on two methods: direct integration for simple kinetics and trail-and-error fitting, which is more commonly used for complex systems. As a rule of thumb, $\mathrm{k_j}$ is chosen so that there is a good fit of $\mathrm{Y_i(t)}$ from the mechanism output with the observed experimental data collected at a set of discrete time points.

In reality, reaction networks may contain hundreds of reactions and thus difficult to determine all the rate constants simultaneously. A more realistic way is to start from sub-components of the whole system. In a CRN context for example, if $n$ species are involved in a set of $R$ reactions, researchers usually start with just one specific reaction with varying amount of sub-components. Their concentrations at different time points were measured and the specific $k$ for this particular reaction was then determined. More components are then individually added back for the determination of whole set of $k$s [3]. However, this method are experimentally and computationally expensive and often hindered by the incomplete knowledge of the system.

### 2.3 Emerging Machine Learning Techniques

CRNs are essentially a set of ODEs $\dot{\mathrm{y(t)}} = \mathrm{f(y(t), \theta)}$, where the rate constants $K$ are represented by ODE parameters $\theta$. As we mentioned above, accurate and efficient ODE parameter estimation remains the bottleneck in a number of application areas, including CRNs, dynamics and control, and system biology. To address this problem, people in machine learning society are developing several models, trying to infer ODE parameters with higher efficiency and accuracy.

A couple of studies have been focusing on Bayesian estimation of parameters. Gradient matching avoids explicit integration by considering an alternative model $y(t) = g(t, \phi)$ [4]. However, the results has proven to be inconsistent with numerical integration sometimes [5]. Recent studies have demonstrated the use of Gaussian process within the gradient matching framework to expedite the computation for better results [5, 6]. However, the experiments were only tested in toy problems such as the Lotka-Volterra model and signal transduction cascade model [7, 8].

In addition, recurrent neural network (RNN) has been recently proposed for modelling different biological networks. People applied RNN to capture the nonlinear behavior of blood glucose metabolism, insulin dynamics, and cell signaling [9, 10, 11]. Those problems are all formulated as ODE systems. Each neuron represents the change of one component in the system. All those studies provide great insight for building machine learning models for CRNs.

## 3 Methods

The determination of reaction constants is the major focus of our work. We propose to apply Gaussian Process and Recurrent Neural Network, using linear regression as a baseline, to automatically learn reaction constants. From literature search, we believe these two are the most promising to learn large CRNs. We first analyzed toy-models such as ozone kinetics and circadian oscillation by KaiC phosphorylation and then extended to large reaction networks of $H_2-O_2$ combustion reactions. Here we provide a brief summary of data generation, machine learning methods, and the evaluation metrics.

### 3.1 Data Generation

For each of the case, our data are a set of observations $Y_i$ at different time points $t$. Ideally, these data should be experimentally collected. For the purpose of this project, we selected well-defined systems from well-known chemical kinetics with known rate constants and generate pseudo-experimental observations from numerical integration, which are served as raw input for our machine learning

algorithms to infer rate constants. Several packages including Scipy [12], Chempy [13] and Sympy [14] are used here.

### 3.2 Linear Regression

Linear Regression (least-square) is used as our base-line method. For this method, we transformed the ODEs into the form of $Ax = b$, where $x$ is a matrix containing the rate constants of interest. The training data is matrix $A$ of size $S(n-1) \times m$, where $S$ represents the number of ODEs that describes $\frac{dY_i}{dt}$, $n$ is the number of observations of each species, and $m$ is the number of rate constants that we want to evaluate.

Note that in real CRNs, parameters of different terms in different ODEs are related by stoichiometry. To avoid treating each parameter individually, we didn't directly feed $\{Y_i$ (observed concentration of species Y), $\frac{dY_i}{dt}\}$ into the built-in linear regression package. Instead, now each row of $A$ contains the possible combination of $\prod Y_i$ according to the form of ODE. If for example $Y_2 * Y_3$ does not appear in the first ODE, the corresponding cell in $A$ is simply 0. The output column vector $b$ has a length of $S(n-1)$, each row representing the derivative of $\frac{dY_i}{dt}$ estimated by $\frac{Y_{i+1}-Y_i}{t_{i+1}-t_i}$. Then we get $x$ from $\arg\min_x ||Ax - b||_2$.

### 3.3 Gaussian Processes (GPs)

Gaussian processes (GPs) is a non-parametric statistical process (a collection of random variables indexed by time or space). The comparison between explicit integration of an ODE system and our GP model used in this project is shown in Fig.2. As we show in the figure, the input from kinetic experiment, $y^*(t)$, are usually contaminated with some measurement error which is modeled as $y^*(t) = y(t) + \epsilon$, where $\epsilon$ is modelled as a zero-mean Gaussian with variance $\sigma^2$. To obtain values for explicit solutions y(t), the ODE system must be solved with an initial condition $y_0$ and ODE parameters, i.e., rate constants K. By employing appropriate priors,

$$p(\mathbf{K}, y_0, \sigma) \propto \pi(\mathbf{K})\pi(y_0)\pi(\sigma) \prod_n \boldsymbol{N}_{Y_n^*}(Y_n(K, Y_0), I\sigma_n^2) \tag{2}$$

the desired marginal $p(K|Y^*)$ can be obtained from this joint posterior. However, every sampling requires integration and a specific solution of the ODE system, which, as explained in Related Work, is the main computational bottleneck.

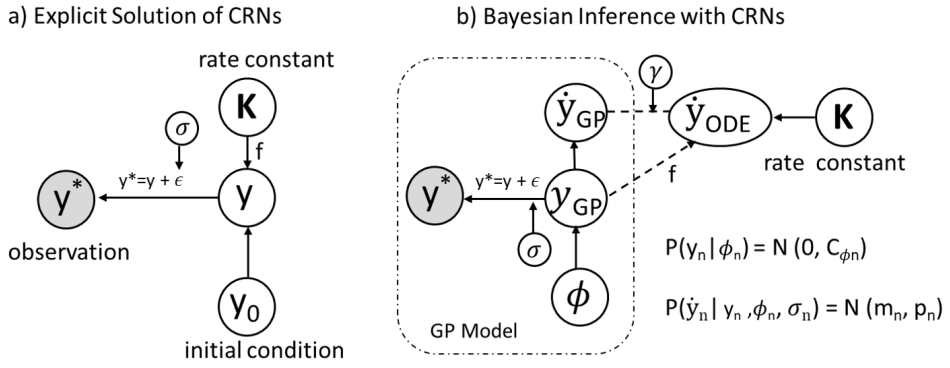



Figure 2: (a) Graphical model of explicit solutions of an ODE system. (b) Graphical model of adaptive gradient matching with Gaussian process

The GP model used in this work follows the previous work [5, 6]. We extended the R package deGradInfer[15]. The basic principles are described here. We assume independent Gaussian process priors on the state variables as $p(Y_n \mid \phi_n) = N(0, C_{\phi_n})$, where $C_{\phi_n}$ denotes the matrix of covariance function (similarity measurements between points, i.e., the kernel function). With noise term as $\epsilon \sim N(0, \sigma^2)$, the state posterior, $p(Yn, | Yn^*, \sigma, \phi)$ follows a normal distribution. The conditional distribution for the state-derivatives as $p(\dot{Y}_{GP} \mid Y_{GP}, \phi, \sigma)$ is also Gaussian. The mean

and variance of this distribution are related with $C_{\phi_n}$, auto covariance for each state derivative and the cross covariances between the state and its derivatives. (See [5] for more details). The GP model in the dashed box in Fig.2(b) specifies a joint Gaussian distribution over the function and its derivatives, which allows us to evaluate a posterior over parameters K consistent with the differential equation f. Normal errors are assumed between the state derivatives $\dot{y}_{GP}$ and $\dot{y}_{ODE}$,where $\dot{y}_{ODE}$ is the functional f(Y, K) evaluated at the GP-generated state values $Y_{GP}$. $\gamma$ is the state-specific error.

Introducing priors as $\pi(K)$ and $\pi(\gamma)$, we can calculate $p(K, \gamma \mid Y, \phi, \sigma)$. The ODE system now is not required to be explicitly solved. GPs can provide a distribution of our estimated parameters. The mean of the posterior distribution, is our MAP estimates of parameters ($k_j$ in our case). People already show that GP approaches can run up to two orders of magnitude faster than the conventional numerical integration [7, 16, 8]. Therefore, it is promising to apply GP models in the reconstruction of CRNs.

### 3.4 Recurrent neural network (RNN)

Recurrent neural network (RNN) is a type of artificial neural network designed to recognize patterns in sequences of data, formulated as $\mathrm{Y_t} = \mathrm{Y_{t-1}} + \mathrm{f(Y_{t-1}, K)}$. The state an RNN reached at prior times will affect the later state. Therefore, RNNs have two sources of input, the present and the recent past, which combined to determine how they respond to new data. One can imagine that if the number of sequential evaluations in RNN is gradually increased, it will eventually converge to an infinitesimal (differential) form as $\mathrm{dY/dt} = \mathrm{f(Y, K)}$, which is exactly in the form of an ODE system. This mathematical link between RNN and ODE implies that we can apply RNNs for dynamical problems [9, 17].

The RNN model used in this work follows the previous work [9]. It is implemented by the deep learning package Keras [18]. The architecture is shown in Fig.3. Y(t) represents the concentration of the chemical species. K represents the rate constants, and f represents the the rate of production (ROP) for individual chemical species in the system. The number of neurons (depicted by the circles with f symbol) is determined by the number of species. Each neuron represents ROP and the corresponding change in concentration within $\Delta t$. The concentration at t + $\Delta t$ is obtained by adding the neuron output to the initial system state. This is fed back as input at the next time step (dashed black lines) and the process is repeated to implement Eq. 1 iteratively.

The RNN model follows a finite difference scheme, which requires the hyper-parameter $\Delta t$ not too large or too small. Large time step will distort the captured nonlinear trends but small time step expends unnecessary computational effort. Another hyperparamter is the learning rate (lr), which controls how much to change each time the model weights are updated. Small lr may result in a long training process whereas large lr may result in learning a sub-optimal set or an oscillated error output. In this work, lr is set between $10^{-4} - 10^{-3}$ to avoid unstable training.

There are several advantages of implementing RNN for CRN reconstruction. First, RNN model is an efficient representation of a chemical reaction network. What's more, the corresponding rate constants of the reactions are represented by weights in the RNN model, making it possible to directly estimate them during training. Besides, it is amenable to train with gradient descent methods to minimize ODE loss between RNN outputs and observations from the data generation process, defined as

$$\mathrm{Loss} = \left\| \mathrm{Y_{output}} - \mathrm{Y_{observation}} \right\|^2 \tag{3}$$

In this work, we applied Adam (adaptive moment estimation) [19] as the optimizer for gradient descent.

## 4 Results and Discussions

We applied linear regression with poly-nominal terms (as our baseline method), gradient matching with GP, and RNN to reconstruct CRNs. We evaluated these machine learning algorithms based on both the robustness of the preservation of system dynamics and the deviation of the inferred ODE parameters, i.e., determined reaction rate constant, from their true values. After an exemplary learning case to validate our learning model with a simple kinetics ($O_2$−$O_3$−O), we looked into

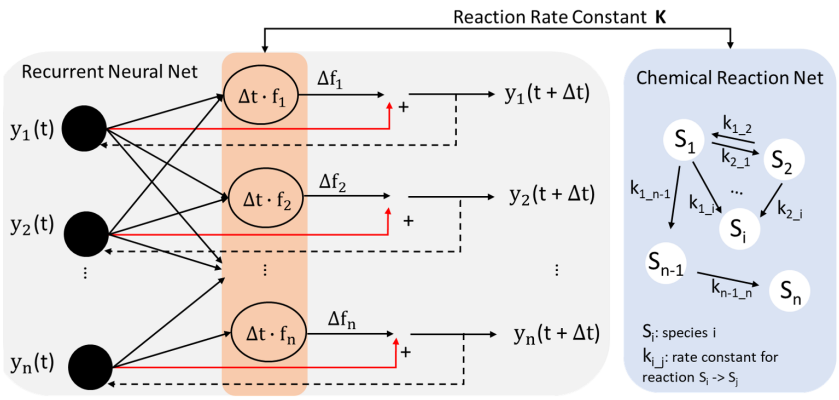
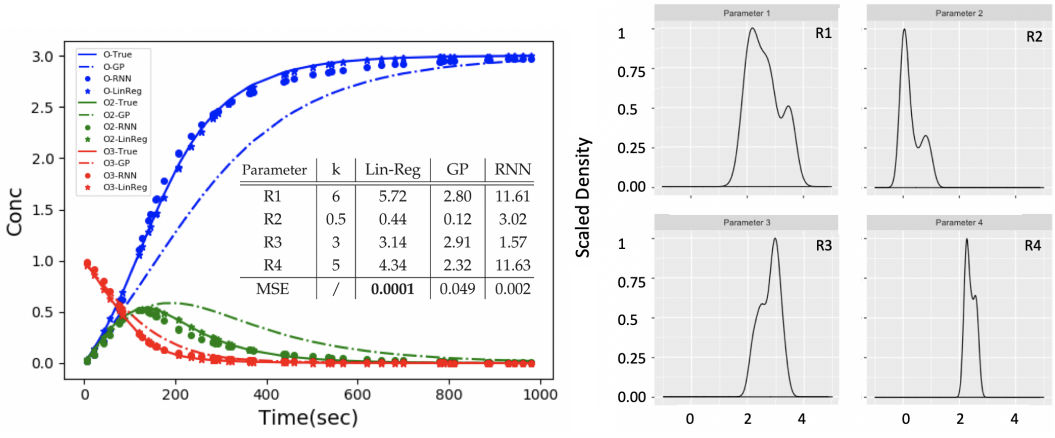



Figure 3: Architecture of the RNN model. The solid arrows represent feed forward. The dotted arrows represent feedback. Reaction rate constants from CRNs control the output from functional f.

real case kinetics: a biochemical kinetic oscillation and $H_2-O_2$ combustion mechanism with 14 reactions.

### 4.1 Case 1: Atmospheric $O_2-O_3-O$ Kinetics

As show in Fig.4, we found that all 3 models performed reasonably well since they captured the overall $O_2-O_3-O$ dynamics. Here we used solid line to present the ground-truth concentration of O, $O_2$, and $O_3$. From Fig.4, we can see the curve generated by the learned parameters from RNN and linear-reg basically overlap with the ground-truth curve, while GP shows some extent of shifts. In least square regression (Lin-reg), we selected 50 time points as our observations, which rendered a training data $A$ of size $147 \times 4$, and a column vector $b$ of length 197 (3 species and 3 ODEs). Selecting 100 random observed time-points, we found linear regression was the best performing model with an MSE of 0.0001. Although all 3 models produced similar dynamics, the parameters learned by linear regression were closest to the true rate constants. GP also gave us a distribution of learned parameters as shown in the right figure.




| Parameter | k | Lin-Reg | GP | RNN |
|---|---|---|---|---|
| R1 | 6 | 5.72 | 2.80 | 11.61 |
| R2 | 0.5 | 0.44 | 0.12 | 3.02 |
| R3 | 3 | 3.14 | 2.91 | 1.57 |
| R4 | 5 | 4.34 | 2.32 | 11.63 |
| MSE | / | **0.0001** | 0.049 | 0.002 |

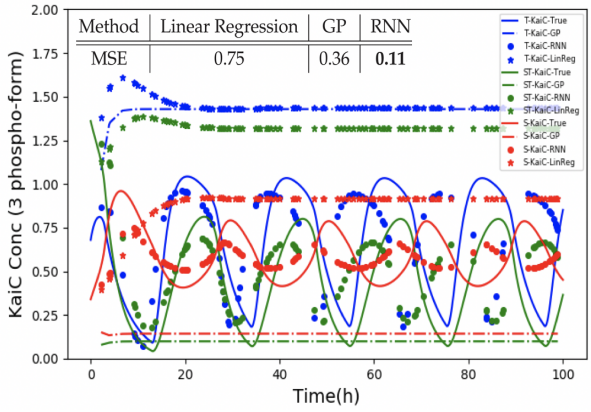
Figure 4: $O_2-O_3-O$ Kinetics Model Performance and Learned GP Parameter Distribution

### 4.2 Case 2: Biochemical Kinetic Oscillation

Here we demonstrate the rate constants learning process with a real 22h-period circadian oscillator found in cyanobacteria. The major feature for this system is the robust 22hr oscillation. We assessed how good our learned parameters by how well the learned parameters could preserve the system

dynamics. As we show in Fig.5, both gradient matching with Gaussian process and linear regression failed to capture the oscillatory dynamics. In contrast, rate constants learned by RNN (represented as circles) regenerated the closest oscillation pattern.

In this case, we have a total of 5 species and 16 parameters (See Related Work for governing equations). In least square regression, we selected 100 time points as our observations, which rendered a training data $A$ of size $297 \times 16$, and $b$ of length 297. As we show here, the modeled systems by GP and lin-reg very quickly reached a steady state where none of the 3 phospho-forms of KaiC continued oscillating. RNN appeared to be our best model with an MSE of 0.11 assessed from 100 randomly selected time points. We observed that the amplitude and phase of each oscillation still shifted a little comparing to the ground-truth, but the period remained approximately 22hr. In addition, the learned rate constants from RNN had the smallest deviation. Due to the complex nature of the system, GP failed to converge within a reasonable amount of time, therefore we only assessed two parameters (running time is 3hr). Yet, even with only 2 parameters to learn, it failed to produce the desirable dynamics (Parameter distribution from GP is not shown here due to its poor performance). We speculate that in order to mimic this oscillatory pattern, we may need to update the current RBF kernel with some periodical kernels for the GP model.




| Rate Constants | True Value | Lin-Reg | GP | RNN |
|---|---|---|---|---|
| $K^0_{UT}$ | 0 | 0.246 | / | 0.0091 |
| $K^0_{TD}$ | 0 | 0.248 | / | 0.016 |
| $K^0_{SD}$ | 0 | 0.071 | / | 0.042 |
| $K^0_{US}$ | 0 | -0.058 | / | 0.048 |
| $K^0_{TU}$ | 0.21 | -0.106 | / | 0.55 |
| $K^0_{TD}$ | 0 | 0.208 | / | 0.0012 |
| $K^0_{DS}$ | 0.31 | 0.113 | 2.773095 | 0.39 |
| $K^0_{SU}$ | 0.11 | 0.092 | 1.958643 | 0.31 |
| $K^A_{UT}$ | 0.479077 | -0.0053 | / | 1.19 |
| $K^A_{TD}$ | 0.212923 | -0.0003 | / | 0.13 |
| $K^A_{SD}$ | 0.505692 | -0.0023 | / | 0.61 |
| $K^A_{US}$ | 0.0532308 | 0.0015 | / | 0.19 |
| $K^A_{TU}$ | 0.0798462 | 0.0053 | / | 0.23 |
| $K^A_{TD}$ | 0.1730000 | 0.00013 | / | 0.18 |
| $K^A_{DS}$ | -0.319385 | 0.0030 | / | -0.51 |
| $K^A_{SU}$ | -0.133077 | -0.00049 | / | -0.088 |

Figure 5: Model Performance of Circadian Oscillator with KaiC Phosphorylation and Inferred Rate Constants

### 4.3 Case 3: Towards Large-Scale Kinetics

Most previous studies are limited to small-scale kinetics [9, 5], i.e., no more than three state variables with up to ten parameters to infer. The performances of machine learning models in large scale ODE systems are not discussed and understood well. As chemical reaction networks are intrinsically large-scale, it is of great priority to implement machine learning algorithms in a more complex CRN. What's more, from previous cases, we observed that linear regression requires special care for CRN problems and performed poorly in nonlinear dynamics. Even though GP avoids direct ODE integration, the computational time for matrix inversion, marginalization is non-negligible (t $\sim$ O(10 h)). RNN best reserved the system dynamics with reasonable computational cost (t $\sim$ O(2 h)). Therefore, we choose RNN model to demonstrate the feasibility of machine learning methods for large-scale CRN reconstruction.

The simulated dynamics for a $H_2$−$O_2$ mechanism at 1 atm and 1000 K is shown in Fig.6. We found that there is a good match between RNN output dynamics and experimental observations. The ground-truth rate constants, deviations from initial guess, and RNN training results are listed in Table 1. From the table, we see that 14 reaction rates differ by orders of magnitude, from O($10^{-13}$) to O($10^{-11}$). We observed that the reaction rate constants of R2, R13, and R14 are trained pretty well by RNN while the rate constants of R4-R9 show no improvement. R1, R11, and R12 even deviate more from the initial guess. Interestingly, well-trained parameters from RNN are mostly in O($10^{-12}$), while extreme rate constants, i.e., 'outliers', (e.g. R3, R9 ) are trained poorly. We postulate that this is caused by the sensitivity of the system, that is, species dynamics may not be sensitive to some reactions given the initial concentration and initial guess of the rate constants.

For example, R9 is extremely fast compared with other reactions. In other words, before any other species dynamics could even start, OH or $HO_2$ is already consumed completely. Consequently there is no chance to learn reaction rate constant of R9 well from the current setting. Similarly, some reactions (e.g. R1) are too slow to have an impact on the system dynamics. Unless there is a tremendous accumulation of $H_2$ or O, R1 does not has a clear impact on the species dynamics. Since those outliers ( $O(10^{-13})$ or $O(10^{-11})$) are difficult to train, some rate constants of $O(10^{-12})$ are further deviated to "compensate" the loss caused by outliers. From the above results, RNN show great capability to preserve CRN dynamics. However, to infer parameters correctly, more sensitivity analysis or model reduction should be performed before any learning-based reconstruction.

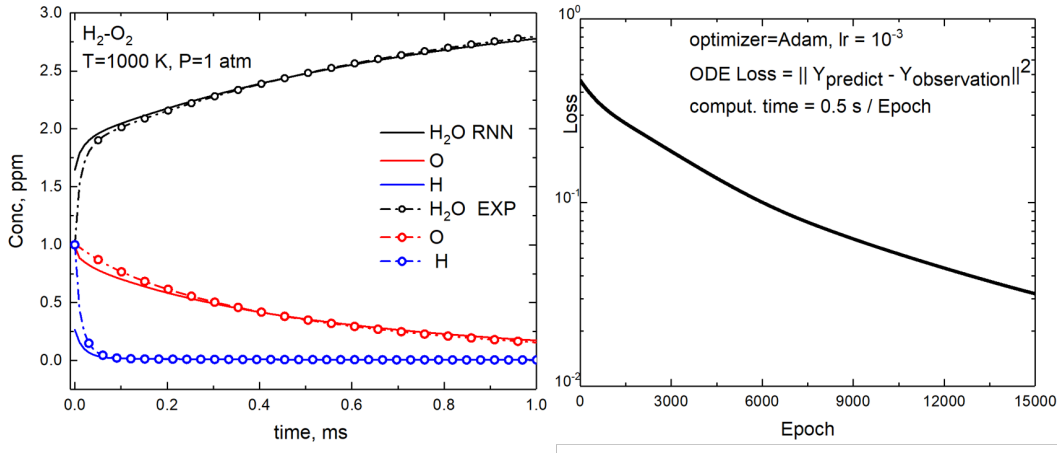



Figure 6: Left: Comparisons of simulated dynamics for H,O,$H_2O$ between RNN inferred rate constants and "experimental observations" generated from Chempy package [13]. Right: Loss decay within 15,000 epochs.

### 4.4 Possible Extensions

There are several extensions from the current work. As we discussed in case 3, large stiffness is a distinctive feature of CRNs. Rate constants may vary up to several orders of magnitudes. Model reduction is necessary before any learning-based reconstruction, that is, we should neglect reactions with relatively small contribution to the system dynamics and therefore simplify the whole network structure. Besides, all rate constants in the current work are assumed to be constant across time. In reality, there is a strong nonlinear dependence between rate constants and temperature, pressure, etc. We need to incorporate Arrhenius law or Troe fitting to consider such dependence. What's more, this study is limited to a fully observed system, i.e,. we assume at observation time, all species concentrations are available. However, in real kinetic experiments, researchers can never measure all species profiles at one time. So how to reconstruct CRN from a partial observed system? That would be a more practical problem for chemical kinetics society.

## 5 Conclusions

In this work we first formulated the chemical reaction network (CRN) and then developed possible machine learning algorithms including linear regression, adaptive gradient matching with Gaussian process (GP), and recurrent neural network (RNN) to infer reaction rate constants, i.e., reconstruct CRNs from discrete experimental observations. We demonstrated those methods with three cases: $O_2$-$O_3$-O kinetics, circadian oscillation by KaiC phosphorylation, and $H_2-O_2$ combustion mechanism. Overall, we found that RNN best reserves the system dynamics from small to large-scale systems. Lin-Reg (with poly-terms) performs best in simple kinetics but fails to capture nonlinear dynamics in larger systems. GP has the worst performance and we postulate that substituting with a periodic kernel may greatly improve its performance. With more extensions including model reduction, incorporation of nonlinear dependence between rate constants and temperature, pressure, etc., this work has great potential for accurate estimation of rate constants and efficient analysis of the dynamic behaviour for large-scale realistic chemical reaction networks (CRNs).

| | Reactions | k | Initial $\epsilon$ | RNN $\epsilon$ |
|---|---|---|---|---|
| R1 | $O + H_2 \longrightarrow H + OH$ | $6.85 \times 10^{-13}$ | 0.06 | 0.10 |
| R2 | $H_2 + OH \longrightarrow H_2O + H$ | $1.37 \times 10^{-12}$ | 0.39 | 0.02 |
| R3 | $O + H_2O \longrightarrow OH + OH$ | $1.37 \times 10^{-13}$ | 0.30 | 0.59 |
| R4 | $H + OH \longrightarrow H_2O$ | $1.37 \times 10^{-10}$ | 0.40 | 0.40 |
| R5 | $H + O_2 \longrightarrow HO_2$ | $1.37 \times 10^{-11}$ | 0.50 | 0.47 |
| R6 | $HO_2 + H \longrightarrow H_2 + O_2$ | $2.74 \times 10^{-11}$ | 0.33 | 0.32 |
| R7 | $HO_2 + H \longrightarrow OH + OH$ | $6.85 \times 10^{-11}$ | 0.58 | 0.59 |
| R8 | $HO_2 + O \longrightarrow O_2 + OH$ | $2.74 \times 10^{-11}$ | 1.64 | 1.62 |
| R9 | $HO_2 + OH \longrightarrow H_2O + O_2$ | $4.11 \times 10^{-11}$ | 1.90 | 1.91 |
| R10 | $2\,HO_2 \longrightarrow H_2O_2 + O_2$ | $2.74 \times 10^{-12}$ | 1.34 | 1.20 |
| R11 | $H_2O_2 + H \longrightarrow H_2O + OH$ | $2.74 \times 10^{-12}$ | 0.34 | 0.48 |
| R12 | $H_2O_2 + H \longrightarrow HO_2 + H_2$ | $1.37 \times 10^{-12}$ | 0.01 | 0.30 |
| R13 | $H_2O_2 + O \longrightarrow OH + HO_2$ | $1.37 \times 10^{-12}$ | 0.32 | 0.03 |
| R14 | $H_2O_2 + OH \longrightarrow HO_2 + H_2O$ | $1.37 \times 10^{-12}$ | 0.23 | 0.08 |

Table 1: This reduced $H_2-O_2$ mechanism is adapted from the well-validated HP Mech developed at Princeton [20]. k ($cm^3$/molecule s) is the reaction rate constant from online NIST chemical kinetics database [21]. $\epsilon$ is the deviation from the true rate constants defined as $\|(\mathrm{k} - \mathrm{k}_{\mathrm{true}})/\mathrm{k}_{\mathrm{true}}\|$

## References


[1] A. Gorban and G. Yablonsky, "Three waves of chemical dynamics," *Mathematical Modelling of Natural Phenomena*, vol. 10, no. 5, pp. 1–5, 2015.

[2] L. Barrie, J. Bottenheim, R. Schnell, P. Crutzen, and R. Rasmussen, "Ozone destruction and photochemical reactions at polar sunrise in the lower arctic atmosphere," *Nature*, vol. 334, no. 6178, p. 138, 1988.

[3] M. J. Rust, J. S. Markson, W. S. Lane, D. S. Fisher, and E. K. O'shea, "Ordered phosphorylation governs oscillation of a three-protein circadian clock," *Science*, vol. 318, no. 5851, pp. 809–812, 2007.

[4] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao, "Parameter estimation for differential equations: a generalized smoothing approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 5, pp. 741–796, 2007.

[5] B. Calderhead, M. Girolami, and N. D. Lawrence, "Accelerating bayesian inference over nonlinear differential equations with gaussian processes," in *Advances in neural information processing systems*, pp. 217–224, 2009.

[6] F. Dondelinger, D. Husmeier, S. Rogers, and M. Filippone, "Ode parameter inference using adaptive gradient matching with gaussian processes," in *Artificial intelligence and statistics*, pp. 216–228, 2013.

[7] D. Barber and Y. Wang, "Gaussian processes for bayesian estimation in ordinary differential equations," in *International Conference on Machine Learning*, pp. 1485–1493, 2014.

[8] M. Niu, S. Rogers, M. Filippone, and D. Husmeier, "Fast inference in nonlinear dynamical systems using gradient matching," *Proceedings of Machine Learning Research*, vol. 48, pp. 1699–1707, 2016.

[9] H. Ling, S. Samarasinghe, and D. Kulasiri, "Novel recurrent neural network for modelling biological networks: oscillatory p53 interaction dynamics," *Biosystems*, vol. 114, no. 3, pp. 191–205, 2013.

[10] R. Albert and R.-S. Wang, "Discrete dynamic modeling of cellular signaling networks," *Methods in enzymology*, vol. 467, pp. 281–306, 2009.

[11] M. A. González-Olvera, A. G. Gallardo-Hernández, Y. Tang, M. C. Revilla-Monsalve, and S. Islas-Andrade, "A discrete-time recurrent neurofuzzy network for black-box modeling of insulin dynamics in diabetic type-1 patients," *International Journal of Neural Systems*, vol. 20, no. 02, pp. 149–158, 2010.

[12] E. Jones, T. Oliphant, and P. Peterson, "{SciPy}: Open source scientific tools for {Python}," 2014.

[13] B. Dahlgren, "Chempy: A package useful for chemistry written in python," *Journal of Open Source Software*, vol. 3, no. 24, p. 565, 2018.

[14] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, *et al.*, "Sympy: symbolic computing in python," *PeerJ Computer Science*, vol. 3, p. e103, 2017.

[15] B. Macdonald and F. Dondelinger, "Ode parameter inference with degradinfer," 2017.

[16] M. Heinonen, C. Yildiz, H. Mannerström, J. Intosalmi, and H. Lähdesmäki, "Learning unknown ode models with gaussian processes," *arXiv preprint arXiv:1803.04303*, 2018.

[17] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems*, pp. 6571–6583, 2018.

[18] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

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

[20] M. P. Burke, M. Chaos, Y. Ju, F. L. Dryer, and S. J. Klippenstein, "Comprehensive h2/o2 kinetic model for high-pressure combustion," *International Journal of Chemical Kinetics*, vol. 44, no. 7, pp. 444–474, 2012.

[21] W. G. Mallard, F. Westley, J. Herron, R. F. Hampson, and D. Frizzell, "Nist chemical kinetics database," *Version 2Q98, National Institute of Standards and Technology, Gaithersburg, MD*, 1998.