

Neural-network solutions to stochastic reaction networks

Ying Tang (✉ jamestang23@gmail.com)

Beijing Normal University <https://orcid.org/0000-0002-9272-8570>

Jiayu Weng

Beijing Normal University

Pan Zhang

Institute of Theoretical Physics, Chinese Academy of Sciences

Article

Keywords:

Posted Date: November 22nd, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-2180974/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: There is **NO** Competing Interest.

Version of Record: A version of this preprint was published at Nature Machine Intelligence on March 16th, 2023. See the published version at <https://doi.org/10.1038/s42256-023-00632-6>.

Neural-network solutions to stochastic reaction networks

Ying Tang,^{1,*} Jiayu Weng,^{2,†} and Pan Zhang^{3,4,5,‡}

¹*International Academic Center of Complex Systems,
Beijing Normal University, Zhuhai 519087, China*

²*Faculty of Arts and Sciences, Beijing Normal University, Zhuhai 519087, China*

³*CAS Key Laboratory for Theoretical Physics, Institute of Theoretical Physics,
Chinese Academy of Sciences, Beijing 100190, China*

⁴*School of Fundamental Physics and Mathematical Sciences,
Hangzhou Institute for Advanced Study, UCAS, Hangzhou 310024, China*

⁵*International Centre for Theoretical Physics Asia-Pacific, Beijing/Hangzhou, China*

The stochastic reaction network is widely used to model stochastic processes in physics, chemistry and biology. However, the size of the state space increases exponentially with the number of species, making it challenging to investigate the time evolution of the chemical master equation for the reaction network. Here, we propose a machine-learning approach using the variational autoregressive network to solve the chemical master equation. The approach is based on the reinforcement learning framework and does not require any data simulated in prior by another method. Different from simulating single trajectories, the proposed approach tracks the time evolution of the joint probability distribution in the state space of species counts, and supports direct sampling of configurations and computing their normalized joint probabilities. We apply the approach to various systems in physics and biology, and demonstrate that it accurately generates the probability distribution over time in the genetic toggle switch, the early life self-replicator, the epidemic model and the intracellular signaling cascade. The variational autoregressive network exhibits a plasticity in representing the multimodal distribution by feedback regulations, cooperates with the conservation law, enables time-dependent reaction rates, and is efficient for high-dimensional reaction networks with allowing a flexible upper count limit. The results suggest a general approach to investigate stochastic reaction networks based on modern machine learning.

I. INTRODUCTION

The stochastic reaction network is a standard model for stochastic processes in physics [1], chemistry [2, 3], biology [4, 5], and ecology [6, 7]. Representative examples include the birth-death process [8], the model of spontaneous asymmetric synthesis [9], and gene regulatory networks [10, 11]. In particular, due to the rapid development of measuring molecules at the single-cell level, it becomes increasingly important to model intracellular reaction networks which have low counts of molecules and are subject to random noise [12]. These stochastic reaction networks in a well-mixed condition can be exactly modeled by the chemical master equation (CME) [13], which describes a time-evolving probability distribution of discrete states representing counts of reaction species. However, the number of possible states grows exponentially with the number of species; thus, it is generally not possible to exactly represent the joint probability and solve the CME.

Many efforts have been made to approximately solve the CME, and a variety of numerical methods have been developed. The most popular method, the Gillespie algorithm [14–18], also known as the stochastic simulation algorithm or the kinetic Monte Carlo, simulates trajectories to generate statistics of relevant variables. It is based on the Monte Carlo method by sampling from all possible paths (or trajectories). However, achieving high-accuracy statistics of the joint probability distribution requires a large number of trajectories. Moreover, the dynamics can be dramatically affected by rare but important trajectories, which are difficult to sample by the Gillespie algorithm [19, 20]. Different from the sampling-based methods, asymptotic approximations have been proposed to transform the CME into continuous-state equations, e.g., the chemical Langevin equation [21, 22]. This method enjoys efficient solutions based on differential equations, but the continuous-state approximation becomes inaccurate when the count of species is small, e.g., in the case of proteins [12]. The stochastic noise originating from the discrete count of species

* These authors contributed equally; Corresponding authors: jamestang23@gmail.com

† These authors contributed equally

‡ Corresponding authors: panzhang@itp.ac.cn

may also induce abrupt changes in stability, known as the noise-induced transition [23–25]. Another class of methods truncates the CME into a state space covering the majority of the probability distribution, including the finite state projection method [26, 27], the sliding window method [28], and the ACME method [29, 30] based on the finite-buffer technique [31]. Further advances employ the Krylov subspace approximation [32] and tensor-train representations [33, 34]. However, the computational cost of these methods is still prohibitive to reach high accuracy when both the number (type) and counts of species become large [35]. Although a great effort has been made, we still lack a general method to solve the CME by directly facing the representation problem of the evolving joint probability of species counts.

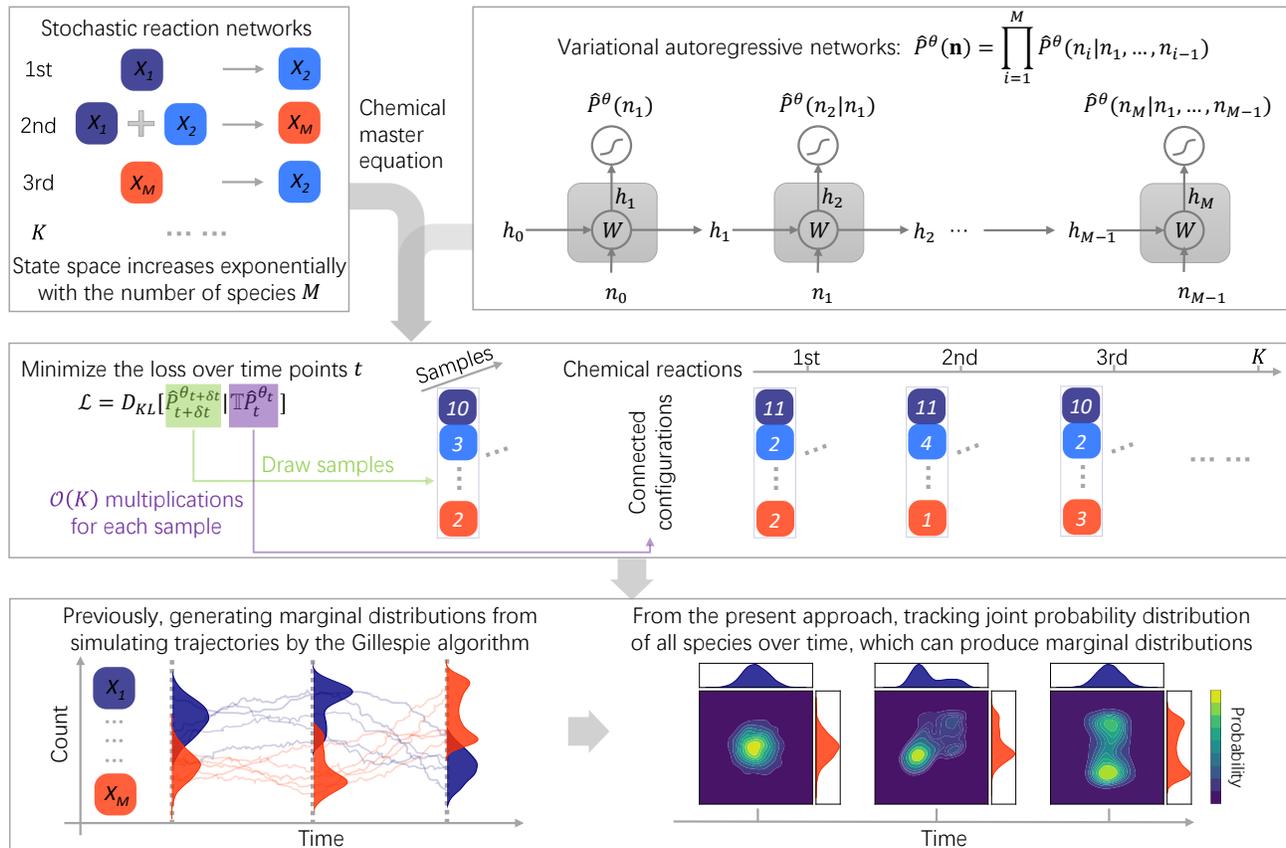


Fig. 1. A schematic of tracking the joint probability distribution over time for stochastic reaction networks. (Upper) For a reaction network, the state space scales exponentially with the number of species M , making it difficult to track the time evolution of the joint distribution. We leverage the variational autoregressive network, such as with the unit of recurrent neural networks, to parameterize the joint count distribution $\hat{P}^\theta(\mathbf{n})$. (Middle) Starting from an initial distribution, we minimize the loss function by the KL-divergence between joint distributions at consecutive time steps to learn its time evolution. The subscript t denotes time points, θ denotes parameters of the neural network to be optimized, and \mathbb{T} is the transition kernel of the chemical master equation. To train the VAN at time $t + \delta t$, samples are drawn from the distribution $\hat{P}^\theta_{t+\delta t}$. Each sample is illustrated by a column of stacked squares with color specifying the species and the count of every species inside the square. For each sample, the number of connected configurations, e.g., those transit into the sample through the transition operator, is equal to the number of chemical reactions K . Thus, only the order of $\mathcal{O}(K)$ multiplications is required to account for the transitions at each time step. (Bottom) The previous method of simulating trajectories by the Gillespie algorithm may generate marginal distributions, but is computationally prohibitive to accurately produce high-dimensional joint distributions. Here, we can track the joint distribution of all species (the two-species are plotted for visualization) by the variational autoregressive network, which allows a flexible upper count limit of species and characterizes high-dimensional joint distributions (see text).

In this paper, we develop a neural-network computational approach to investigate the time evolution of the joint probability distribution for the stochastic reaction network. Our method is inspired by the strong representation power and the efficient sampling ability of autoregressive neural networks for high-dimensional data [36, 37]. In particular, we leverage the variational autoregressive network (VAN) to solve the CME (Fig. 1). The VAN has been applied to statistical physics [38, 39], quantum-many body systems [40–42], open quantum systems [43, 44], and computational biology [45, 46], where it enables directly sampling configurations, computing the normalized

probability of configurations, and estimating macroscopic thermodynamical quantities including the free energy and entropy. Here, we extend the VAN to characterize the joint probability distribution of species counts for the stochastic reaction network. As the unit of the VAN, we have employed recurrent neural networks (RNN) [40] and the transformer [47], which are flexible to represent the high-dimensional probability distribution and to adjust the upper count limit. Furthermore, we enable the VAN to have the count constraint on each species separately if a priori knowledge is available or to maintain the conservation on the total count of species for specific systems, both of which can improve the accuracy with the contracted probability space.

The present approach differs significantly from the recently developed methods. Instead of using the simulated data from the Gillespie algorithm to train the neural network [48, 49], our approach does not need the aid of any existing data or measurements. This is advantageous, especially for cases where the Gillespie algorithm itself is insufficient to capture the multimodal distribution [20]. The present approach results in an automatically normalized distribution as the solution of the CME at arbitrary finite time, different from learning the transition kernels [50, 51] or approximating only the steady state [52]. The obtained joint distribution contains more information than estimating the marginal statistics alone [35], providing the probability for each configuration in the high-dimensional state space.

To demonstrate the advances of the proposed approach, we apply it to representative examples covering typical stochastic reaction networks in physics, chemistry and biology: the genetic toggle switch with multimodal distribution [19], the early life self-replicator with an intrinsic constraint of count conservation [7], the epidemic model with time-dependent rates [53], and the intracellular signaling cascade in the high dimension [35]. The results on the marginal statistics match those from the previous numerical methods, such as the Gillespie algorithm or the finite state projection [26, 27]. The present approach also efficiently produces the time evolution of the joint probability distribution, as visualized by the joint distribution for any two of the species. In particular, it can generate a multimodal distribution consistent with the ACME method [29, 30], is effective for systems with feedback regulations, and is computationally efficient for high-dimensional systems. For example, the computational time scales sublinearly to the number of species in the signaling cascade with the possible count of species around $\mathcal{O}(10^2)$. We provide a code repository, which only requires to input the stoichiometric matrix, initial conditions, and reaction rates, allowing the approach to be applicable and adaptable to more stochastic reaction networks.

A. Chemical master equation

We consider the discrete-state continuous-time Markovian dynamics with configurations $\mathbf{n} = \{n_1, n_2, \dots, n_M\}$ and the size of variables M . The probability distribution at time t evolves under the stochastic master equation [8]:

$$\partial_t P_t(\mathbf{n}) = \sum_{\mathbf{n}' \neq \mathbf{n}} W_{\mathbf{n}' \rightarrow \mathbf{n}} P_t(\mathbf{n}') - r_{\mathbf{n}} P_t(\mathbf{n}), \quad (1)$$

where $P_t(\mathbf{n})$ is the probability of the system being in a configuration state \mathbf{n} , $W_{\mathbf{n} \rightarrow \mathbf{n}'}$ are the transition rates from configuration \mathbf{n} to \mathbf{n}' , and the escape rate from \mathbf{n} is $r_{\mathbf{n}} = \sum_{\mathbf{n}' \neq \mathbf{n}} W_{\mathbf{n} \rightarrow \mathbf{n}'}$. As the probability of transiting into and out from any configuration sum up to 0, the probability following this equation is conserved over time.

One type of the master equation describing stochastic reaction networks is the chemical master equation. The chemical reactions are given by the stoichiometric matrix and reaction rates. Specifically, a stochastic reaction network with K reactions and M species (each species of the molecule X_i has the count $n_i = 0, 1, \dots, N_i$ with $1 \leq i \leq M$) is:



with the reaction rate constants c_k for the k -th reaction. The numbers of reactants and products are denoted by r_{ki} and p_{ki} , respectively. The change in species under the chemical reactions is given by the stoichiometric matrix: $s_{ki} = p_{ki} - r_{ki}$.

To formulate the CME for a given stochastic reaction network, a set of propensity functions need to be specified. The propensities, $a_k(\mathbf{n})$, represent the probability of the reaction k occurring in the infinitesimal time interval at the state \mathbf{n} [2]. For example, according to the law of mass action [11], the propensity a_k for each reaction is calculated by multiplying the rate constant c_k and the number of chemical species X_i by the power of r_{ki} . In general, the propensities can have nonlinear terms of the numbers of species.

Given the stoichiometric matrix, rate constants, propensities and an initial distribution $P_0(\mathbf{n})$, the system evolves according to the chemical master equation [13]:

$$\partial_t P_t(\mathbf{n}) = \sum_{k=1}^K [a_k(\mathbf{n} - s_k) P_t(\mathbf{n} - s_k) - a_k(\mathbf{n}) P_t(\mathbf{n})], \quad (3)$$

where s_k is the k -th row of the stoichiometric matrix. We consider the reflecting boundary condition for both boundaries at $n_i = 0$ and $n_i = N_i$, by setting the transition probability out of the boundary to zero. Then, the CME conserves the probability over time, as in the present examples. Other boundary conditions such as the absorbing boundary can be implemented when necessary. Then, the CME may not conserve the probability, but the VAN can still learn an automatically normalized distribution and give the normalization constant [39].

The state space of the probability distribution scales exponentially with the number of species: N^M when each species has a count up to N . The exponentially increasing state space makes it challenging to solve the CME and track the time evolution of the probability distribution. Based on a neural-network ansatz, we next provide a computational approach toward solving this problem.

II. FORMULATION

A. Design of variational autoregressive networks

We leverage the variational autoregressive network (VAN) [38] to represent the probability distribution of species counts. The VAN is a product of conditional probability for each configuration:

$$\hat{P}^\theta(\mathbf{n}) = \prod_{i=1}^M \hat{P}^\theta(n_i | n_1, \dots, n_{i-1}), \quad (4)$$

where n_i is the count of the i -th chemical species, and θ denotes the parameters of the VAN to be trained. The VAN parameterizes an automatically normalized probability distribution, such that sampling configurations can be performed in parallel under the corresponding probabilities from the distribution.

The conventional unit of the VAN includes MADE [54], PixelCNN [55], a recurrent neural network (RNN) [40] and a transformer [47]. Here, we mainly employ an RNN and a transformer to represent the highly correlated distributions. In the examples below, we find that a single-layer RNN is sufficient to parameterize the joint probability distribution, and the transformer performs equally well. Details on the architecture of the RNN and transformer are provided in the Methods.

Compared with the VAN for statistical mechanics [38], where each variable has binary states for the spin, we extend it to the N -state variable. We choose N as the maximum of the upper count limit of all the species for a given system: $N = \max_i N_i$. If N is too large, training the VAN may require a longer computational time to converge. When a priori knowledge of the upper count limit N is unavailable for a given system, one can first find the smallest upper count limit N , such as by simulating trajectories from the Gillespie algorithm. Another more rigorous approach to finding the proper N is to use the ACME method [29, 30]. This method based on the finite-buffer method [31] can reduce the state space from a M -dimensional hypercube to a M -simplex, and provides an error bound on choosing N . It gives guidance on choosing an appropriate N to truncate, ensuring that the truncation error is smaller than a threshold. After finding a proper N , the flexibility of neural networks in representing complex distributions allows N to be adjusted accordingly in the present method.

For specific systems, there may be a priori knowledge of the count of each species, of an intrinsic conservation on the total count of species. These constraints can be put into the VAN to improve the accuracy of training in the contracted probability space. The details of such implementations are given as follows.

1. Adding constraints on the count of each species

When a priori knowledge on the count of certain chemical species is available, one can put separate constraints on the count of each species, to reduce the size of the state space and improve the accuracy of the VAN. For example, the DNA fragment typically has a count of zero or one inside a cell. Thus, one can constrain its count to 0, 1, which reduces the state space of the probability distribution.

We implement this constraint in the VAN, by setting the probability of the exceeding count in the output of the VAN unit to zero. That is, the count of the species out of the constraint is set to zero probability, such that only the allowable count is generated in the VAN. The constraint on the count can be specified for each chemical species, as an input argument in our code package. It is exemplified in the toggle switch and the autoregulatory feedback loop.

2. Keeping conservation on the total species count

Many stochastic reaction networks in nature include a count conservation of chemical species. Such conservation places a constraint on the total count of certain chemical species. Therefore, count conservation needs to be incorporated into the VAN. Here, we implement the total count conservation in the VAN similar to as [40, 42], to model the chemical reactions with an intrinsic conserved quantity N_0 . Such an implementation of count conservation will be used in the example of an early life self-replicator.

To make the total count conserved, we adjust the output of each conditional probability in the VAN, to assign zero probabilities to the count of species that do not satisfy the conservation. Specifically, the conditional probabilities $\hat{P}^\theta(n_i|n_1, \dots, n_{i-1})$ for species i are zero if

$$0 \leq n_i \leq N_0 - \sum_{j=1}^{i-1} n_j \quad (5)$$

is not satisfied. This constraint ensures that the sampled configurations from the VAN must have the total count of species equal to N_0 . We have used such strategies of adjusting the neural-network output for both sampling and estimating the probability from the VAN. The conservation of certain species can be realized in a similar manner.

We note that conservation can also be maintained by performing the variable reduction. However, putting the constraint on the total conservation without the variable reduction is more straightforward in the VAN. Specifically, for a high-dimensional system with many species under a constraint of $\sum_{i=1}^M n_i = N_0$, after performing the variable reduction, the remaining variables are still under the constraint of their count and are dependent on the count of other species.

B. Tracking time evolution of the probability distribution

To track the probability distribution over time, we minimize the loss function:

$$\mathcal{L} = D_{KL}[\hat{P}_{t+\delta t}^{\theta_{t+\delta t}} | \mathbb{T} \hat{P}_t^{\theta_t}], \quad (6)$$

which is the Kullback-Leibler (KL)-divergence between a distribution $\hat{P}_{t+\delta t}^{\theta_{t+\delta t}}$ parameterized by the VAN and the one-time-step evolved distribution $\hat{P}_t^{\theta_t}$ at time t . The transition kernel in the time interval δt is $\mathbb{T} = e^{\delta t \mathbb{W}} \approx (I + \delta t \mathbb{W})$, which preserves the total probability. The generator \mathbb{W} is given by Eq. (1) for the system under study. By design, the VAN parameterizes an automatically normalized probability distribution [38] at each time step. The time step length dt is typically fixed, but an adaptive dt can be used to help accelerate the time evolution (Methods).

The loss function is approximated by drawing samples from the VAN:

$$\mathcal{L} = \mathbb{E}_{\mathbf{s} \sim \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}} \{ \ln \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}(\mathbf{s}) - \ln[(\mathbb{T} \hat{P}_t^{\theta_t})(\mathbf{s})] \}, \quad (7)$$

where \mathbf{s} are samples drawn from the distribution $\hat{P}_{t+\delta t}^{\theta_{t+\delta t}}$ parameterized by the VAN. At each time step, we can only evaluate the manipulation of the configurations transiting in and out of each drawn sample. The number of connected configurations for each sample in the CME is equal to the number of chemical reactions K . Thus, we only need to perform the order of $\mathcal{O}(K)$ multiplications to account for all the possible transitions of each configuration at each time step, rather than performing $\mathcal{O}(N^M)$ multiplications of the full transition matrix that increase exponentially with the number of species M . These manipulations are also paralleled for each sample. This crucial property enables us to approximately solve the chemical master equation by the VAN.

The VAN can be updated by the REINFORCEMENT approach [56]:

$$\nabla_{\theta_{t+\delta t}} \mathcal{L} = \mathbb{E}_{\mathbf{s} \sim \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}} \{ [\nabla_{\theta_{t+\delta t}} \ln \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}(\mathbf{s})] \cdot \{ \ln \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}(\mathbf{s}) - \ln[(\mathbb{T} \hat{P}_t^{\theta_t})(\mathbf{s})] \} \}, \quad (8)$$

where we have used $\mathbb{E}_{\mathbf{s} \sim \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}} [\nabla_{\theta_{t+\delta t}} \ln \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}(\mathbf{s})] = \nabla_{\theta_{t+\delta t}} \sum_{\mathbf{s}} \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}(\mathbf{s}) = \nabla_{\theta_{t+\delta t}} 1 = 0$ when sufficient samples are drawn. Variance reduction is included to reduce the variance of the loss function [38]. The pseudocode is summarized in ALGORITHM 1.

Although training the VAN requires sampling, the VAN represents the joint distribution covering the full state space, which is essentially different from sampling-based methods such as the Gillespie algorithm. After learning the distribution at each time step with a sufficiently small step length, dynamics at separated time scales can be revealed.

Input: System dimension, the stoichiometric matrix, reaction rates, the rule of propensities, the initial distribution, and hyperparameters (the upper count limit, time steps, step length, size of the VAN, learning rate, batch size, the number of epochs).

Output: The joint probability distributions over time and relevant statistics.

Initialize the distribution, choose hyperparameters of the VAN.

for Every time step **do**

Learn the next-step VAN $\hat{P}_{t+\delta t}^{\theta_{t+\delta t}}$:

for Every epoch **do**

1. Draw samples from VAN, $\mathbf{s} \sim \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}$.
2. Generate connected configurations, which transit from or into the sample \mathbf{s} through the chemical reactions.
3. Acquire the probabilities of the connected configurations from the current-step VAN $\hat{P}_t^{\theta_t}$.
4. Calculate the matrix elements of the transition kernel \mathbb{T} for the connected configurations.
5. Train the VAN $\hat{P}_{t+\delta t}^{\theta_{t+\delta t}}$ by minimizing the loss function: $\mathcal{L} = \mathbb{E}_{\mathbf{s} \sim \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}} \{\ln \hat{P}_{t+\delta t}^{\theta_{t+\delta t}}(\mathbf{s}) - \ln[(\mathbb{T} \hat{P}_t^{\theta_t})(\mathbf{s})]\}$.
6. Save the VAN $\hat{P}_{t+\delta t}^{\theta_{t+\delta t}}$ for the next time step.

end for

Calculate statistics from the joint distribution $\hat{P}_{t+\delta t}^{\theta_{t+\delta t}}$ for each time step.

end for

Search for optimal hyperparameters, by comparing with marginal statistics from other numerical methods (optional).

ALGORITHM 1. Solving the chemical master equation by the VAN. The rule of propensities can be specified such as by the conventional law of mass action [11]. For the choice of hyperparameters, the upper count limit is the upper bound on the maximum count of all species and should be chosen properly for the given system, such as by the Gillespie simulation or the ACME method [29, 30]. The initial choice of hyperparameters can be similar to those in Supplementary Table II.

III. RESULTS

We next apply our approach to representative examples of stochastic reaction networks, including the genetic toggle switch [19], the early life self-replicator [7], the epidemic model [53], and the intracellular signaling cascade [35]. They separately demonstrate that our approach is applicable to systems with a multimodal distribution, with an intrinsic constraint of count conservation, with time-dependent parameters, and in high dimensions. More examples are provided in the Supplementary: the birth-death process (Supplementary Fig. 1), gene expression without regulation (Supplementary Fig. 2), an autoregulatory feedback loop (Supplementary Fig. 3), and more cases of the intracellular signaling cascade. The details of the VAN used for each model are listed in Supplementary Table II.

To evaluate the accuracy of the learnt distribution, we compare the resultant marginal distribution of one species with those from the other numerical methods, including the Gillespie algorithm or the finite state projection. For the present models, the Gillespie simulation is sufficient for the comparison of the marginal statistics, based on the analytical or numerical analysis in the original literature of the models. The finite state projection is used as a comparison in the autoregulatory feedback loop model, where the bimodal distribution of the protein is captured consistently by this method and the VAN. The distance between the marginal distributions of our method and the other numerical method is quantified by the Hellinger distance. We also find a match between the VAN and the ACME method on learning a multimodal distribution for the genetic toggle switch. To further highlight the flexibility of the VAN to capture the full joint probability distribution, in each example we show the joint distribution for two of the species (heatmaps in each figure), as the joint distribution of more species cannot be visualized clearly. With the VAN, tracking the joint distribution becomes feasible and computationally efficient for high-dimensional systems.

A. Toggle switch

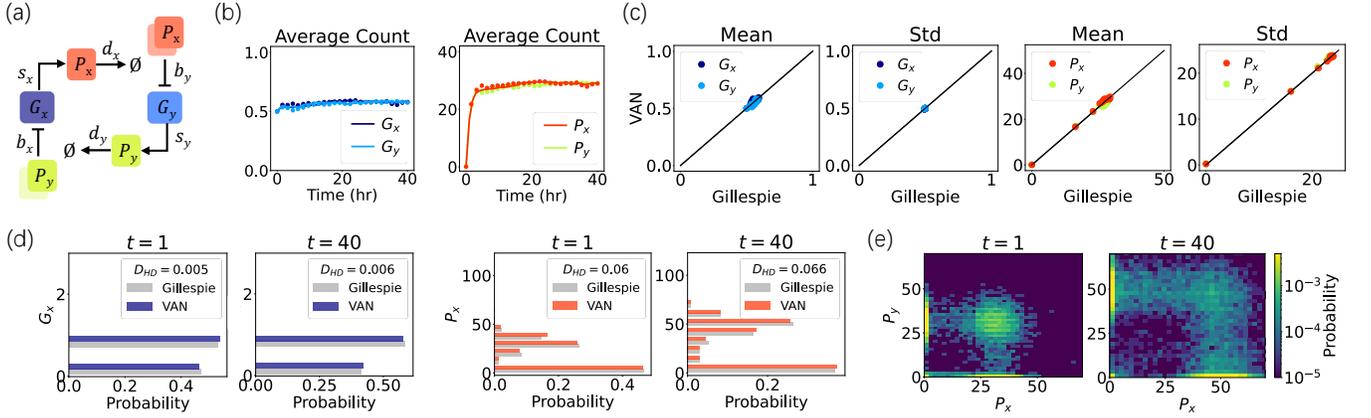
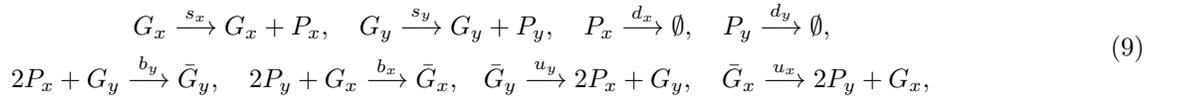


Fig. 2. The result of the genetic toggle switch. (a) A schematic of the reactions. (b) The time evolution of the average count for the genes and proteins, from the VAN (dots) and the Gillespie simulation (lines). (c) Comparison of the means and standard deviations of the counts of the genes and proteins between the VAN and the Gillespie simulation, at time points $t = 0, 1, \dots, 39, 40$. (d) The marginal distributions of the gene G_x and protein P_x at time points $t = 1, 40$ from the Gillespie simulation (gray) and the VAN with the same color in (b). The inset shows the Hellinger distance between the two distributions. (e) The joint distribution of the two proteins from the VAN at time points $t = 1, 40$, with the color bar for the probability values in the logarithmic scale. In the long-time limit, there are four stable states with probability peaks located at $(P_x, P_y) = \{(0, 0), (50, 0), (0, 50), (50, 50)\}$, corresponding to the four genetic states of the two genes: “Off-Off”, “On-Off”, “Off-On”, and “On-On”. The Gillespie simulation has $4 * 10^3$ trajectories. The initial distribution has an equal count of the bounded and unbounded forms for each gene and zero count of the proteins. The hyperparameters of the VAN are in Supplementary Table II, and the parameters are $s_x = s_y = 50$, $d_x = d_y = 1$, $b_x = b_y = 10^{-4}$, and $u_x = u_y = 0.1$.

We consider the genetic toggle switch [19, 57] as illustrated in Fig. 2a. This system has a multimodal distribution on the protein counts, depending on the genetic states of the two mutually inhibited genes. Thus, the system is suitable for testing the flexibility of the VAN to learn the multimodal distribution.

There are six molecular species: genes G_x, G_y , which transcript two proteins P_x, P_y inhibiting the gene expression of each other. There are two protein-DNA complexes \bar{G}_x, \bar{G}_y , with protein P_y (P_x) bound to gene G_x (G_y). The dimer of the protein P_x (P_y) inhibits the activity of the gene G_y (G_x). The model can be written as the chemical reactions:



where s_x, s_y are the synthesis rates of the proteins, and p_x, p_y are the degradation rates of the proteins. The transition rate b_y (b_x) is the binding rate of two copies of protein P_x (P_y) to the G_y (G_x), to form the complex \bar{G}_y (\bar{G}_x). The unbinding of the complex \bar{G}_y (\bar{G}_x) has the rate u_y (u_x).

The total count of the two forms for each gene is conserved: $G_x + \bar{G}_x = 1$. This conservation sets a constraint on the count of the two genes: $G_x = 0, 1$, $G_y = 0, 1$, and effectively reduces two variables by $\bar{G}_x = 1 - G_x$, $\bar{G}_y = 1 - G_y$. We have put this constraint on the count of the two genes in the neural network of the VAN. We next consider the parameter regime with weak promoter binding, because in this regime the joint distribution is multimodal with four probability peaks, posing a challenge to accurately track the joint distribution.

The result from the VAN matches those from the Gillespie simulation, including the mean (Fig. 2b) and standard deviation (Fig. 2c). The marginal distributions (Fig. 2d) on the count of the gene G_x and protein G_y are also consistent between the two methods. The joint distribution of the two proteins (Fig. 2e) shows the multimodality corresponding to the four stable genetic states of the two genes: “Off-Off” ($G_x = 0, G_y = 0$), “On-Off” ($G_x = 1, G_y = 0$), “Off-On” ($G_x = 0, G_y = 1$), and “On-On” ($G_x = 1, G_y = 1$), which is consistent with the probability distribution from the ACME method shown in Fig. 4 of [19]. The result demonstrates that the VAN can accurately produce the multimodal probability distribution.

B. Early life self-replicator

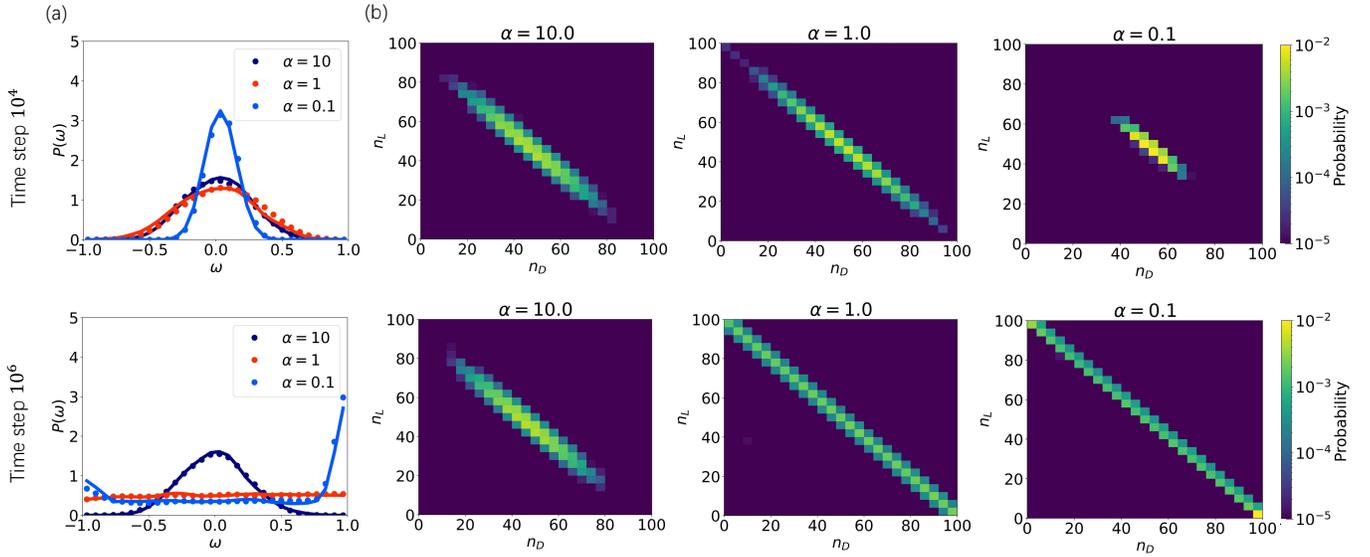
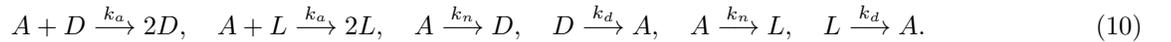


Fig. 3. The result of the early life self-replicator. (a) The order parameter $\omega = (n_D - n_L)/(n_D + n_L)$ from the VAN (dots) and the Gillespie simulation (line), at the time steps 10^4 and 10^6 , where the latter reaches close to the steady state. The values of the critical parameter $\alpha_c = 10, 1, 0.1$ separately lead to unimodal, flat, and bimodal distributions in the long-time regime, as denoted by the color. (b) The joint distribution of D, L at time steps 10^4 and 10^6 . The long-time joint distribution also shows unimodal, flat, and bimodal shapes (higher probabilities at the corner), where the color bar denotes the probability values in the logarithmic scale. The parameters are $k_a = k_n = k_d = 1$, and $\alpha = V k_n / k_a$ with the volume V . The Gillespie simulation has 10^3 trajectories. The hyperparameters of the VAN are in Supplementary Table II, and the initial distribution is the delta distribution at $n_A = 3, n_D = 50, n_L = 47$ with the total count $N_0 = 100$. The time step length for the VAN is $dt = 4 \times 10^{-3}, 4 \times 10^{-4}, 4 \times 10^{-5}$ for $\alpha = 10, 1, 0.1$, as the three cases have different time scales.

We consider a chemical reaction modeling the biological homochirality of an early life self-replicator [7], which is an extension of the original model of spontaneous asymmetric synthesis [9]. One property of the model is that the species intrinsically has a total count conservation. To maintain count conservation, we used Eq. (5) in the implementation of the VAN, which effectively learns the distribution with conservation.

The system contains three species A, D, L , six reactions and three rate constants. The enantiomers D, L of a chiral molecule are produced autocatalytically from an achiral molecule A . It has the following chemical reactions.



The rate constants k_a, k_n, k_d have the subscript identify the specific reaction. Another parameter $\alpha = V k_n / k_a$ is introduced, corresponding to the volume V in which the reactions occur.

The state of this system is described by a chiral order parameter $\omega = (n_D - n_L)/(n_D + n_L)$ from the counts of D, L . The bifurcation on the stability at the steady state was found, which is controlled by the parameter α . The steady-state distribution is given analytically by $P_{ss}(\omega) = \mathcal{N}(1 - \omega^2)^{\alpha-1}$, where $\mathcal{N} = \Gamma(\alpha + 1/2)/[\sqrt{\pi}\Gamma(\alpha)]$ and V denotes the volume [7]. At the critical value of the parameter α_c , the system changes between a unimodal system with the order parameter $\omega = 0$ in the racemic state, and a bimodal system with $\omega = \pm 1$ in the homochiral states, which was confirmed by the Gillespie simulation. The bimodal distribution was found to have two probability peaks concentrated on the two boundaries of the order parameter, $\omega = \pm 1$.

We consider three cases of the critical parameter: $\alpha_c = 10, 1, 0.1$ as above, equal, and below the critical value $\alpha_c \approx 1$. In the long-time regime, the probability distribution $P(\omega)$ by the VAN has unimodal and bimodal shapes for $\alpha_c = 10, 0.1$ separately (Fig. 3a), which matches the Gillespie simulation. The bimodal distribution has a higher peak at the right side of the boundary, because the initial distribution has more $n_D = 50$ than $n_L = 47$, and will converge to the steady state with an equal height of two peaks after longer simulations. The two sharp peaks at the boundary $\omega = \pm 1$ are sensitive to the collected samples for performing the histogram, which affects both the Gillespie simulation and the VAN approach. Increasing the batch samples in the VAN and the trajectories in the Gillespie simulation can lead to a more accurate estimation. Different from simulating trajectories by the Gillespie algorithm, the present approach

generates the time evolution of the joint probability distribution in the state space of D, L (Fig. 3b). Starting from a unimodal distribution, the VAN approach can reveal the emergence of unimodal and bimodal joint distributions over time.

C. Epidemic model

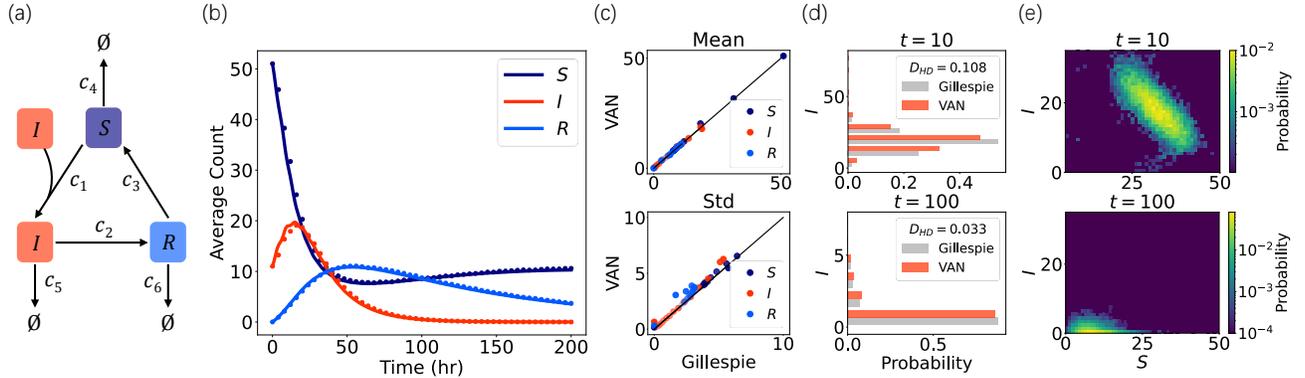
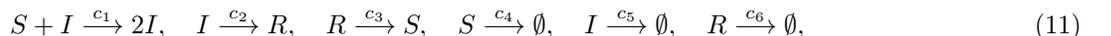


Fig. 4. The result of the epidemic model. (a) A schematic of the chemical reaction. (b) The time series of the average count of species from the VAN (dots) and the Gillespie simulation (lines). The color specifies the chemical species. (c) Comparison of the means and standard deviations of the chemical species between the VAN and the Gillespie simulation, at time $t = 0, 10, \dots, 190, 200$ for chemical species specified by color. (d) The marginal count distributions of various chemical species are plotted horizontally at time $t = 10, 100$. The color in (d) specifies the result from the VAN for the species I , and gray denotes the Gillespie method. The inset contains the Hellinger distance between the two distributions. (e) The joint distribution of S and I at time $t = 10, 100$ from the VAN, with the color bar denoting the probability values in the logarithmic scale. The values of the parameters: $c_0 = 0.003$, $\omega = \pi/3$, $\epsilon = 0.2$, $c_2 = 0.02$, $c_3 = 0.007$, $c_4 = 0.002$, $c_5 = 0.05$ and $c_6 = 0.002$, with units of per hour [53]. The Gillespie simulation has 10^4 trajectories. The initial distribution is the delta distribution with $S = 51$, $I = 11$ and $R = 0$.

We consider the epidemic SIR model with a time-dependent contact rate [53], which helps demonstrate the effectiveness of our method in the system with time-dependent rates. The system models the transmission of an infectious disease, and contains three species: susceptible S , infected I and recovered R . A susceptible population S can be infected when it comes into contact with infected population I . The infected population I may become the recovered population R , which can become susceptible again. Each population dies at a certain rate.

A schematic of the chemical reaction is illustrated in Fig. 4a. It has the following chemical reactions:



where c_1 is a time-dependent contact rate modeled as a periodic function as $c_1 = c_0(1 + \epsilon) \sin(\omega t)$ and the remaining parameters are constant. This time-dependent rate models the infection in a periodic environment.

We apply the present approach to estimate the evolution of joint probability distribution over time. As a comparison, the average count of the three species estimated from the VAN match with the Gillespie simulation (Fig. 4b). The means, the standard deviations (Fig. 4c) and marginal distributions (Fig. 4d) of the two methods are consistent with each other. We further show the joint probability distribution of S and I and its evolution over time points (Fig. 4e). The results demonstrate that the VAN method can accurately track the joint probability distribution for systems with time-dependent rates. Thus, it is applicable to more realistic situations where the parameters vary over time.

D. Intracellular signaling cascade

The intracellular signaling cascade [35] is a biological reaction with a series of reactions, where one species catalyzes the production of the next. This recurrent structure of the signaling cascade makes it an ideal model to conveniently test our method with different numbers of species. Following [35], we consider three cases: (1) a linear signaling cascade without feedback regulation; (2) a nonlinear signaling cascade without feedback regulation; and (3) a linear signaling cascade with feedback regulation.

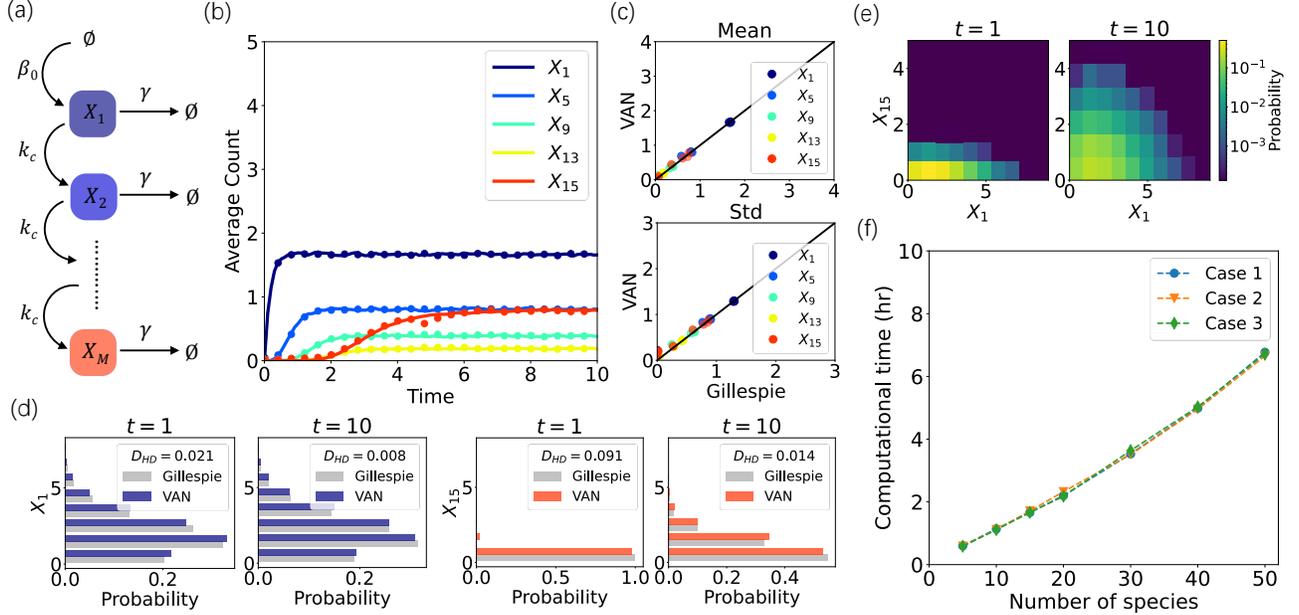
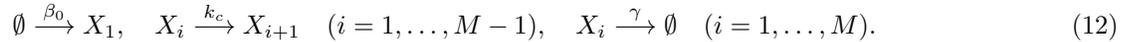


Fig. 5. The result of the linear signaling cascade. (a) A schematic of the chemical reaction. (b) The time series of the average count of species from the VAN (dots) and the Gillespie simulation (lines). The color specifies the chemical species. (c) Comparison of the means and standard deviations of the chemical species between the VAN and the Gillespie simulation, at time points $t = 1, 2, \dots, 9, 10$ for chemical species specified by color. (d) The marginal count distributions of various chemical species are plotted horizontally at time points $t = 1, 10$. The color in (d) specifies the result from the VAN and gray denotes the Gillespie method. The inset contains the Hellinger distance between the two distributions. (e) The joint distribution of X_1 and X_{15} at time points $t = 1, 10$ from the VAN, with the color bar denoting the probability values in the logarithmic scale. Panels (a-e) have $M = 15$. The other parameters are $\beta_0 = 10$, $k_c = 5$, $\gamma = 1$, and the Gillespie simulation has 10^4 trajectories, and the initial distribution is the delta distribution with all species zero. The results for $M = 50$ or cases 2, 3 are in the Supplementary Information. The upper count limit of the species is $N = 100$ for the case 2, and $N = 10$ for cases 1, 3. (f) The computational times for three cases (color and symbol) of the signaling cascade with $M = 5, 10, 15, 20, 30, 40, 50$. The computational times are counted for time step $dt = 0.01$ and time steps 1000. The hyperparameters of the VAN are in Supplementary Table II.

The first case contains chemical reactions:



The parameter β_0 is the synthesis rate, k_c is the catalytic rate, and γ is the decay rate. A schematic of the reaction is shown in Fig. 5a.

The time evolution of the mean species count from the VAN and the Gillespie simulation match with each other (Fig. 5b). The comparison of the means and standard deviations between the two methods also demonstrates that the VAN is accurate for various species and time points (Fig. 5c). We further show the marginal distributions (Fig. 5d), where the small Hellinger distance indicates a similarity between the two distributions from the two methods. The accuracy of the VAN can be further improved by using more samples in the batch and training the neural networks for more epochs.

The above comparison shows a match of the marginal statistics from the two methods. In addition to generating the marginal statistics [35], the VAN gives a joint probability distribution of all the species. In particular, the number of states in the joint probability distribution increases exponentially with the number of species, N^M . Such a joint probability distribution for large M is difficult to capture by simulating trajectories from the Gillespie simulation, but becomes accessible from the VAN. As an illustration, we plot the joint distributions of the first and the last species (Fig. 5e). The application of our method to the system with different numbers of species is given in the Supplementary Information.

We recorded the computational cost versus the size for the three signaling-cascade models (Fig. 5f), with the number of species 5, 10, 15, 20, 30, 40, 50, where the result of $M = 50$ species is shown in Supplementary Fig. 4. The computational time of using the VAN scales sublinearly to the system size (the number of species). Since the number of reactions is

proportional to the number of species, the computational time also scales sublinearly to the number of reactions for this example. In addition, different from the Gillespie algorithm which simulates trajectories to provide the marginal statistics, the VAN approximates the full joint probability distribution under an affordable computational time.

The accuracy is not sensitive to the order of the species used for the conditional probabilities in Eq. (4). Specifically, we have switched the order of species, by starting the signaling from the last species X_M that activates X_1 and so on. The result has the same high accuracy as that of the original order (Supplementary Fig. 5). In addition, the result by the transformer has a similar accuracy compared with the RNN in this example (Supplementary Fig. 6). We have considered a set of transformer hyperparameters (Supplementary Table III), where the hyperparameters include the number of features, the number of heads, the number of encoder and decoder layers, and the dimension of the feed-forward network. The results show a similar accuracy in generating the distribution, but using the transformer requires a longer computational time compared with the RNN.

As in [35], we also consider the case 2 with the nonlinear activation and the case 3 with feedback regulations. The results in Supplementary Figs. 7, 8 show a match between the Gillespie simulation and the VAN method. Thus, the VAN is accurate for stochastic reaction networks with nonlinear interactions and feedback regulations, which are ubiquitous in biological systems [11]. Overall, these results demonstrate that the present approach allows an efficient estimation of the joint probability distribution when the number of species increases, allowing its application to high-dimensional reaction networks.

IV. DISCUSSION

We have developed a computational approach to solve the chemical master equation of stochastic reaction networks, by extending the variational autoregressive networks for the joint probability distribution of species counts. Specifically, the possible state of each species can be a hundred or a thousand, beyond the binary state for each spin in statistical physics [38]. During the time evolution, the connected configurations to each sample depend on the chemical reactions, in contrast to the nonequilibrium systems with given rules for spin flips [39]. These differences require a specific architecture of the VAN and rules of the time evolution. We have achieved such an extension, leading to the first approach of using the neural network alone to solve the chemical master equation. In light of the strong representation power of the neural network [36, 37, 58, 59], the approach is generally applicable to stochastic reaction networks. We have applied it to representative examples in biophysics, systems biology and epidemics, where it shows a high accuracy to track the joint probability distribution over time. The computational cost scales sublinearly to the number of species as indicated in the examples, enabling its application to higher-dimensional reaction networks.

The VAN maps each count of species to the conditional probability of the species conditioned on the previous species. This procedure assigns a specific order to the species. Once the VAN can be trained to achieve zero KL-divergence, the joint distribution is faithfully learned regardless of the species' order used in the conditional probabilities, because the parameters of the VAN are optimized to learn the joint probability distribution instead of each conditional probability. Specifically, we used different orders of the species for the conditional probabilities, such as shuffling the order of species in the example of the signaling cascade, and found that the learned probability distribution was not affected by the chosen order (Supplementary Fig. 5). If the chosen order respects the conditional dependence of species in the given reaction network, the training of the VAN may be more efficient.

As in the example of the early life self-replicator and autoregulatory feedback loop, the VAN can effectively parameterize the bimodal distribution. The VAN also captures the multimodality in the toggle switch [19]. In addition to these examples, the method of annealing is helpful to learn the multimodal distribution, such as the temperature annealing in the Hopfield model on lattices [38] and the variational annealing [60]. Furthermore, sampling rare configurations may be necessary to capture the probability peaks with relatively lower probabilities. An importance sampling can be leveraged for this task [61, 62]. To efficiently sample rare configurations, importance sampling needs to be designed for specific reaction networks, based on the available a priori knowledge on rare configurations. Then, the VAN is capable of learning the rugged probability distribution [39].

Compared with the Gillespie algorithm [14–18] of simulating trajectories, the present approach learns the joint probability distribution. Since the state space increases exponentially with the number of species, it is not feasible to extract the joint probability distribution from simulating trajectories for high-dimensional systems, for example, certain configurations may not be accessed unless an exponentially increasing number of trajectories are simulated. Instead, the joint probability distribution can be learned by the VAN, which offers direct sampling from the joint distribution to compute statistics for any configuration, beyond acquiring only the marginal statistics from the Gillespie algorithm. Moreover, the trajectory space increases exponentially with the number of time points, making it harder to characterize the time evolution of the joint distribution by simulating trajectories and capturing rare trajectories. The

present approach overcomes this issue by evolving the VAN to track the probability distribution over each time point. With the learnt time-evolved distribution covering the rare configurations, the dynamics under time-scale separations including rare trajectories may be revealed.

The computational cost of the finite state projection [26, 27] and the sliding window method [28] increases exponentially with the number of species, because the states need to be explicitly labeled to cast the CME into ordinary differential equations. The ACME method [29, 31] also requires the buffers to be sufficiently large for accurate estimations, which may become computationally expensive when the upper count limit N is large. Instead, the present approach captures the joint probability distribution of the whole state space with an adjustable N , and is efficient when both the upper count limit N and the number of species M are large, such as in the example of a signaling cascade. Given such a property, one can further conduct an estimation on choosing the proper N to control the truncation error [30], and then N can be adjusted accordingly for the neural network without losing the flexibility on learning the joint distribution. In addition, compared with the method based on the tensor network [33, 34], the advantages of the VAN mainly include its generality. The neural-network ansatz is more flexible representing complex probability distributions. For example, the VAN is useful in higher-dimensional lattice systems [39], whereas the matrix product states are mainly for one-dimensional lattices [63].

With the joint probability distribution parameterized by the VAN, evaluating the macroscopic thermodynamical quantities becomes feasible. The partition function, free energy, and entropy can be estimated from the process of learning the VAN [38, 39]. Then, the free energy difference between local minima in the state space enables the analysis of the dynamics of crossing the free energy barrier, the quantification of the transition rates between the free energy local minima, and the control of the stochastic reaction network. Thus, the VAN approach makes a connection from tracking the joint distribution of microscopic states to investigating the macroscopic thermodynamics. Future works also include adapting the present approach to reaction networks in more complex scenarios, such as beyond the well-mixed condition or the Markovian dynamics [48]. To this end, we have developed a user-friendly package, to investigate stochastic reaction networks in more general situations by the neural-network approach.

ACKNOWLEDGMENTS

We thank Jing Liu for sharing the code of the transformer. We acknowledge Jie Liang, Mustafa Khammash, Ankit Gupta, Alexander Hoffmann, Ali Farhat, Farid Manuchehrfar and Online Club Nanothermodynamica for helpful discussions. This work is supported by Project 11747601, 11975294, 12105014 of National Natural Science Foundation of China. P.Z. acknowledges the Key Research Program of Frontier Sciences, CAS Grant No. QYZDB-SSW-SYS032. The HPC is supported by Dawning Information Industry Corporation Ltd and Interdisciplinary Intelligence Super-Computer Center of Beijing Normal University, Zhuhai.

AUTHOR CONTRIBUTIONS

Y.T., P.Z. had the original idea for this work, Y.T. and J.W. performed the study, and all authors contributed to the preparation of the manuscript.

ADDITIONAL INFORMATION

Competing interests: The authors declare no competing interests.

Reprints and permission: information is available online at [URL will be inserted by publisher].

Data availability: The authors declare that the data supporting this study are available within the paper.

Code availability: A pytorch implementation of the present algorithm is included as a supplementary file of the manuscript, and has been deposited at <https://github.com/jamestang23/NNCME>, which will be publicly available as of the date of publication.

V. METHODS

A. Variational autoregressive networks

To efficiently represent the count space by the least possible variables in the VAN, we used the count of each species in the range of $[0, N]$ as the input of the VAN. Alternatively, one may also use the one-hot representation or binary representation for encoding the count of chemical species. For the binary representation, each count of a species is encoded as the binary-digit representation. When using the VAN to sample configurations, the VAN maps binary digits back to N -output probabilities for each species. The required number of input variables for binary encoding increases logarithmically with the maximum count of species.

We next present two architectures to represent the probability Eq. (4) in the VAN, the recurrent neural network and the transformer.

1. Recurrent neural network

In the recurrent neural network (RNN), the conditional probability is calculated iteratively over species, with each site on the chain representing a chemical species. Each site has two inputs: a visible variable with the count of the chemical species and a hidden state. The iteration process starts from an initial variable n_0 and hidden state h_0 , which are chosen as zero. Then, a recurrent cell processes the information from the previous input variable n_{i-1} and hidden state h_{i-1} . It generates a new hidden state h_i , and passes the information of h_i to the next cell.

The recurrent cell also provides the conditional probability of this site $\hat{P}^\theta(n_i|n_1, \dots, n_{i-1})$ with the parameter θ . The conditional probability is obtained from the hidden state, which is acted on a linear transform and a softmax operator $\hat{P}^\theta(n_i|n_1, \dots, n_{i-1}) = \text{Softmax}(Wh_i + b)$. The softmax operation ensures the normalized condition for the probability of all the possible counts of species. The joint probability distribution is obtained by multiplying the iteratively generated conditional probabilities.

Sampling from the probability distribution occurs in a similar manner. To sample configurations with discrete counts, we use the multinomial distribution to generate integers based on the given probability. Given an initial variable and hidden state, the variable n_1 is sampled from the estimated conditional probability. The procedure is repeated to the last species, generating a configuration with a certain count of each species.

To learn the distribution with long-range correlations, we use a gated recurrent unit (GRU) [64] as the recurrent cell. It avoids the vanishing gradient problem for the vanilla recurrent neural network, and is more efficient than the cell of the long-short time memory. The GRU takes the d_h -dimensional hidden state h_i as an input. It updates by the gates:

$$z_i = \sigma(W_{zn}n_{i-1} + W_{zh}h_{i-1} + b_z), \quad (13)$$

$$r_i = \sigma(W_{rn}n_{i-1} + W_{rh}h_{i-1} + b_r), \quad (14)$$

$$\hat{h}_i = \tanh(W_{hn}n_{i-1} + W_{hh}(r_i \odot h_{i-1}) + b_h), \quad (15)$$

$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \hat{h}_i, \quad (16)$$

where W s are the weight matrices, σ denotes the sigmoid activation function, b s are bias vectors, and \odot is the Hadamard product. A reset gate r_i and an update gate z_i interpolate between the previous input hidden state and a candidate hidden state, where the reset gate controls the extent of forgetting on the previous hidden state.

To increase the representative capacity of the VAN, many hidden layers of RNN may be employed. We find that the one-layer RNN is sufficient to generate accurate probability distributions in examples.

2. Transformer

The transformer [47] employs the attention mechanism to learn the correlation between relevant configurations in a distribution. For the conventional transformer model, given a ‘‘query’’, it uses ‘‘key’’ to estimate a correlation to certain ‘‘value’’, and then generates the attention value. We implement the transformer as a unit of the VAN, where the input is the count of each species and the output is the conditional probabilities in Eq. (4) with the normalization property.

Specifically, the transformer has an input layer, where a linear transform and the ReLU function act on the input configurations. Then, positional embedding is applied to the configurations, and the attention matrix for the configurations is calculated subsequently by using multihead attention [47]. These are passed to a linear transform and then a softmax operator as in the RNN. The architecture ensures that the output conditional probabilities are normalized.

For the transformer, the hyperparameters include the number of features in the encoder and decoder inputs d_{model} , the number of heads in the multihead attention models n_{heads} , the number of encoder and decoder layers n_{layers} , and the dimension of the feed-forward network model d_{ff} . The hyperparameter n_{layers} plays a similar role as the number of depths in the RNN, and the other hyperparameters play the role of width. The chosen values of hyperparameters are listed in Supplementary Table III.

We have tested the transformer in the examples of the signaling cascade, where it shows a similar accuracy as the RNN. However, the transformer generally needs a longer computational time than the RNN (Supplementary Table III). Therefore, we have mainly used the RNN for other examples. The representative capacity of the transformer allows its application to more complex reaction networks.

B. Training details

Starting from a chosen initial distribution, we iteratively conduct the minimization by Eq. (6), generating a sequence of the VAN to continuously track the time evolution of the distribution. The first time step requires a long epoch ($\mathcal{O}(10^3)$) to have a converged loss. Then, the learned VAN at the current time step is inherited to train the VAN after the next time-step evolution by Eq. (6). After the first time step, the inherited VAN loss soon converges, because the change in the distribution is generally not dramatic. This inheritance significantly accelerates the training, taking $\mathcal{O}(10^2)$ epochs at the time points after the first time step.

The batch size is chosen as 1000 for each epoch. To estimate the statistics from the distribution more accurately, after the loss converges, we save the batch from the last 20% of the epochs. This collection of batches $1000 \times 100 \times 20\%$ (batch size, epoch at each time point, the chosen percent of the epochs) gives more accurate estimations. One can also save the VAN to generate more samples as desired.

The Adam optimizer [65] is chosen to perform the stochastic gradient descent. We tested learning rates 10^{-5} , 10^{-4} , 10^{-3} , and 10^{-2} , which affect the accuracy of training. We find that 10^{-3} leads to relatively lower loss and better accuracy of training. Certain schedulers for the learning rate can be designed, to help alleviate general optimization issues such as trapping into local minima.

The accuracy of the VAN depends on the number of parameters in the neural network. For the RNN, we have used 2, 4, 8, 16, 32, 64, 128 as the number of hidden states, and found that the number larger than or equal to 16 performs well for the one-layer RNN. A larger number of hidden states and hidden layers may lead to better accuracy, with a cost of longer computational time.

We list the number of depths and widths of the RNN with the best performance under our attempt for each model (Supplementary Table II). The table contains the computational times in various systems with different numbers of chemical species and reactions, under the chosen time-step length. We also list the number of parameters for the transformer, which has equal performance but a longer training time (Supplementary Table III).

C. The choice of time step length

The choice of the time-step length requires care. It has the discretization error from the Suzuki-Trotter decomposition [66] in Eq. (6). An improper choice of a time-step length that is too large may generate negative probabilities due to the escape rate in Eq. (1). A sufficiently small time-step length is ideal for the accuracy, with a cost of longer computational time. Thus, the range of time-step length dt needs to be adjusted to each given system, because the reaction rates have various values for each system. One needs to search for the proper dt by trial and error. For example, in the present gene-expression system, the time step length is often chosen as $dt = 0.02$.

In addition to the fixed time-step length, one may use the adaptive step length at each time step to save computational time. Specifically, by trial and error, we first find a sufficiently small step length as a baseline for each step of time evolution. Then, we increase the baseline value by 100-fold. If it generates negative probabilities, the increase is reduced to 100/2-fold, etc., until reaching the baseline of step length which does not give negative probabilities. This

strategy allows us to explore the largest possible step length, to accelerate the simulation.

-
- [1] M. F. Weber and E. Frey, Master equations and the theory of stochastic path integrals, *Rep. Prog. Phys.* **80**, 046601 (2017).
 - [2] D. T. Gillespie, Stochastic simulation of chemical kinetics, *Annu. Rev. Phys. Chem.* **58**, 35 (2007).
 - [3] H. Ge, M. Qian, and H. Qian, Stochastic theory of nonequilibrium steady states. part ii: Applications in chemical biophysics, *Phys. Rep.* **510**, 87 (2012).
 - [4] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, Stochastic gene expression in a single cell, *Science* **297**, 1183 (2002).
 - [5] M. Assaf, E. Roberts, Z. Luthey-Schulten, and N. Goldenfeld, Extrinsic noise driven phenotype switching in a self-regulating gene, *Phys. Rev. Lett.* **111**, 058102 (2013).
 - [6] R. A. Blythe and A. J. McKane, Stochastic models of evolution in genetics, ecology and linguistics, *J. Stat. Mech.* **2007**, P07018 (2007).
 - [7] F. Jafarpour, T. Biancalani, and N. Goldenfeld, Noise-induced mechanism for biological homochirality of early life self-replicators, *Phys. Rev. Lett.* **115**, 158101 (2015).
 - [8] C. W. Gardiner, *Handbook of Stochastic Methods*, 3rd ed. (Springer-Verlag, Berlin, 2004).
 - [9] F. C. Frank, On spontaneous asymmetric synthesis, *Biochim. Biophys. Acta* **11**, 459 (1953).
 - [10] U. Alon, *An introduction to systems biology: design principles of biological circuits* (CRC press, USA, 2006).
 - [11] P. C. Bressloff, *Stochastic Processes in Cell Biology*, Vol. 41 (Springer, Berlin, 2014).
 - [12] A. Raj and A. van Oudenaarden, Nature, nurture, or chance: stochastic gene expression and its consequences, *Cell* **135**, 216 (2008).
 - [13] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry* (Elsevier, New York, 2007).
 - [14] J. L. Doob, Topics in the theory of markoff chains, *Trans. Am. Math. Soc.* **52**, 37 (1942).
 - [15] D. T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, *J. Comput. Phys.* **22**, 403 (1976).
 - [16] M. A. Gibson and J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, *J. Phys. Chem. A* **104**, 1876 (2000).
 - [17] Y. Cao, D. T. Gillespie, and L. R. Petzold, Efficient step size selection for the tau-leaping simulation method, *J. Chem. Phys.* **124**, 044109 (2006).
 - [18] E. Weinan, T. Li, and E. Vanden-Eijnden, *Applied Stochastic Analysis*, Vol. 199 (American Mathematical Society, 2021).
 - [19] A. Terebus, C. Liu, and J. Liang, Discrete and continuous models of probability flux of switching dynamics: Uncovering stochastic oscillations in a toggle-switch system, *J. Chem. Phys.* **151**, 185104 (2019).
 - [20] A. Terebus, F. Manuchehrfar, Y. Cao, and J. Liang, Exact probability landscapes of stochastic phenotype switching in feed-forward loops: Phase diagrams of multimodality, *Front. Genet.* **12**, 645640 (2021).
 - [21] D. T. Gillespie, The chemical langevin equation, *J. Chem. Phys.* **113**, 297 (2000).
 - [22] S. N. Ethier and T. G. Kurtz, *Markov Processes: Characterization and Convergence* (New York: Wiley-Interscience, 2005).
 - [23] Y. Tang, R. Yuan, G. Wang, X. Zhu, and P. Ao, Potential landscape of high dimensional nonlinear stochastic dynamics with large noise, *Sci. Rep.* **7**, 1 (2017).
 - [24] M. Assaf, E. Roberts, and Z. Luthey-Schulten, Determining the stability of genetic switches: Explicitly accounting for mrna noise, *Phys. Rev. Lett.* **106**, 248102 (2011).
 - [25] W. Horsthemke and R. Lefever, *Noise-Induced Transitions: Theory and Applications in Physics, Chemistry, and Biology*, 2nd ed. (Springer-Verlag, Berlin, 2006).
 - [26] B. Munsky and M. Khammash, The finite state projection algorithm for the solution of the chemical master equation, *J. Chem. Phys.* **124**, 044104 (2006).
 - [27] A. Gupta, J. Mikelson, and M. Khammash, A finite state projection algorithm for the stationary solution of the chemical master equation, *J. Chem. Phys.* **147**, 154101 (2017).
 - [28] T. A. Henzinger, M. Mateescu, and V. Wolf, Sliding window abstraction for infinite markov chains, in *Computer Aided Verification* (Springer, 2009) pp. 337–352.
 - [29] Y. Cao, A. Terebus, and J. Liang, Accurate chemical master equation solution using multi-finite buffers, *Multiscale Model. Simul.* **14**, 923 (2016).
 - [30] Y. Cao, A. Terebus, and J. Liang, State space truncation with quantified errors for accurate solutions to discrete chemical master equation, *Bull. Math. Biol.* **78**, 617 (2016).
 - [31] Y. Cao and J. Liang, Optimal enumeration of state space of finitely buffered stochastic molecular networks and exact computation of steady state landscape probability, *BMC Syst. Biol.* **2**, 1 (2008).
 - [32] S. MacNamara, K. Burrage, and R. B. Sidje, Multiscale modeling of chemical kinetics via the master equation, *Multiscale Model. Simul.* **6**, 1146 (2008).
 - [33] V. Kazeev, M. Khammash, M. Nip, and C. Schwab, Direct solution of the chemical master equation using quantized tensor trains, *PLoS Comput. Biol.* **10**, e1003359 (2014).
 - [34] I. G. Ion, C. Wildner, D. Loukrezis, H. Koepl, and H. De Gerssem, Tensor-train approximation of the chemical master equation and its application for parameter inference, *J. Chem. Phys.* **155**, 034102 (2021).

- [35] A. Gupta, C. Schwab, and M. Khammash, Deepcme: A deep learning framework for computing solution statistics of the chemical master equation, *PLoS Comput. Biol.* **17**, e1009623 (2021).
- [36] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, *Phys. Rep.* (2019).
- [37] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [38] D. Wu, L. Wang, and P. Zhang, Solving statistical mechanics using variational autoregressive networks, *Phys. Rev. Lett.* **122**, 080602 (2019).
- [39] Y. Tang, J. Liu, J. Zhang, and P. Zhang, Solving nonequilibrium statistical mechanics by evolving autoregressive neural networks, *arXiv:2208.08266* (2022).
- [40] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, Recurrent neural network wave functions, *Phys. Rev. Research* **2**, 023358 (2020).
- [41] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, Deep autoregressive models for the efficient variational simulation of many-body quantum systems, *Phys. Rev. Lett.* **124**, 020503 (2020).
- [42] T. D. Barrett, A. Malyshev, and A. Lvovsky, Autoregressive neural-network wavefunctions for ab initio quantum chemistry, *Nat. Mach. Intell.* **4**, 351 (2022).
- [43] M. Reh, M. Schmitt, and M. Gärttner, Time-dependent variational principle for open quantum systems with artificial neural networks, *Phys. Rev. Lett.* **127**, 230501 (2021).
- [44] D. Luo, Z. Chen, J. Carrasquilla, and B. K. Clark, Autoregressive neural network for simulating open quantum systems via a probabilistic formulation, *Phys. Rev. Lett.* **128**, 090501 (2022).
- [45] J.-E. Shin, A. J. Riesselman, A. W. Kollasch, C. McMahon, E. Simon, C. Sander, A. Manglik, A. C. Kruse, and D. S. Marks, Protein design and variant prediction using autoregressive generative models, *Nat. Commun.* **12**, 2403 (2021).
- [46] I. Biazzo, A. Braunstein, L. Dall’Asta, and F. Mazza, Epidemic inference through generative neural networks, *arXiv:2111.03383* (2021).
- [47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, Attention is all you need, in *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017).
- [48] Q. Jiang, X. Fu, S. Yan, R. Li, W. Du, Z. Cao, F. Qian, and R. Grima, Neural network aided approximation and parameter inference of non-markovian models of gene expression, *Nat. Commun.* **12**, 1 (2021).
- [49] A. Sukys, K. Öcal, and R. Grima, Approximating solutions of the chemical master equation using neural networks, *iScience* **25**, 105010 (2022).
- [50] L. Bortolussi and L. Palmieri, Deep abstractions of chemical reaction networks, in *Computational Methods in Systems Biology* (Springer, 2018) pp. 21–38.
- [51] D. Repin and T. Petrov, Automated deep abstractions for stochastic chemical reaction networks, *Inf. Comput.* **281**, 104788 (2021).
- [52] G. Gorin, M. Carilli, T. Chari, and L. Pachter, Spectral neural approximations for models of transcriptional dynamics, *bioRxiv* 2022.06.16.496448 (2022).
- [53] V. H. Thanh and C. Priami, Simulation of biochemical reactions with time-dependent rates by the rejection-based algorithm, *J. Chem. Phys.* **143**, 054104 (2015).
- [54] M. Germain, K. Gregor, I. Murray, and H. Larochelle, Made: Masked autoencoder for distribution estimation, *arXiv:1502.03509* (2015).
- [55] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, Pixel recurrent neural networks, in *International conference on machine learning* (PMLR, 2016) pp. 1747–1756.
- [56] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, *Machine Learning* **8**, 229 (1992).
- [57] T. S. Gardner, C. R. Cantor, and J. J. Collins, Construction of a genetic toggle switch in escherichia coli, *Nature* **403**, 339 (2000).
- [58] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *nature* **521**, 436 (2015).
- [59] Y. Tang and A. Hoffmann, Quantifying information of intracellular signaling: progress with machine learning, *Rep. Prog. Phys.* **85**, 086602 (2022).
- [60] M. Hibat-Allah, E. M. Inack, R. Wiersema, R. G. Melko, and J. Carrasquilla, Variational neural annealing, *Nat. Mach. Intell.* **3**, 952 (2021).
- [61] H. Kuwahara and I. Mura, An efficient and exact stochastic simulation method to analyze rare events in biochemical systems, *J. Chem. Phys.* **129**, 10B619 (2008).
- [62] G. M. Rotskoff, A. R. Mitchell, and E. Vanden-Eijnden, Active importance sampling for variational objectives dominated by rare events: Consequences for optimization and generalization, *arXiv:2008.06334* (2021).
- [63] L. Causer, M. C. Bañuls, and J. P. Garrahan, Finite time large deviations via matrix product states, *Phys. Rev. Lett.* **128**, 090605 (2022).
- [64] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, *arXiv:1409.1259* (2014).
- [65] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv:1412.6980* (2014).
- [66] M. Suzuki, Generalized trotter’s formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems, *Commun. Math. Phys.* **51**, 183 (1976).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SINeuralnetworksolutionstostochasticreactionnetworks.pdf](#)