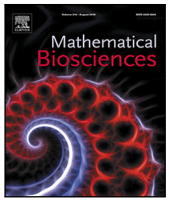
# Efficient and scalable prediction of stochastic reaction–diffusion processes using graph neural networks

Mathematical Biosciences 375 (2024) 109248

Contents lists available at ScienceDirect

Mathematical Biosciences

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



Original Research Article

# Efficient and scalable prediction of stochastic reaction–diffusion processes using graph neural networks

Zhixing Cao [a,c], Rui Chen [d], Libin Xu [a], Xinyi Zhou [a], Xiaoming Fu [a], Weimin Zhong [b], Ramon Grima [e,*]

[a] State Key Laboratory of Bioreactor Engineering, East China University of Science and Technology, Shanghai 200237, China
[b] Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China
[c] Department of Chemical Engineering, Queen's University, Kingston, Canada K7L 3N6
[d] Shanghai Jiao Tong University School of Medicine, Shanghai 200127, China
[e] School of Biological Sciences, the University of Edinburgh, Max Born Crescent, Edinburgh, EH9 3BF, Scotland, United Kingdom

ARTICLE INFO

Dataset link: https://zenodo.org/records/7678839

*Keywords:*
Reaction–diffusion
Master equation
Neural networks

ABSTRACT

The dynamics of locally interacting particles that are distributed in space give rise to a multitude of complex behaviours. However the simulation of reaction–diffusion processes which model such systems is highly computationally expensive, the cost increasing rapidly with the size of space. Here, we devise a graph neural network based approach that uses cheap Monte Carlo simulations of reaction–diffusion processes in a small space to cast predictions of the dynamics of the same processes in a much larger and complex space, including spaces modelled by networks with heterogeneous topology. By applying the method to two biological examples, we show that it leads to accurate results in a small fraction of the computation time of standard stochastic simulation methods. The scalability and accuracy of the method suggest it is a promising approach for studying reaction–diffusion processes in complex spatial domains such as those modelling biochemical reactions, population evolution and epidemic spreading.

## 1. Introduction

A variety of intricate behaviours originate from the dynamics of particles that interact locally and are distributed spatially. These examples exist at all spatial scales, ranging from the microscopic interactions of molecules inside living cells to ecological and epidemic phenomena on a national or continental level [1–6]. The dynamics of these interactions are strongly influenced by spatial heterogeneity and stochasticity, which in turn determine the emergence of collective behaviour.

Stochastic reaction–diffusion processes are useful mathematical models for systems that display complex spatio-temporal behaviours [7–12]. Commonly used approaches model these processes at the individual particle level [13–16]. These frameworks whilst straightforward to implement and intuitive, they are not easily amenable to an analytic treatment and stochastic simulations based on them are also very time consuming if one is interested in understanding stochastic population dynamics. A different approach which avoids these issues involves discretising space into a finite number of volumes (compartments) and using a reaction–diffusion master equation (RDME [17–20]) to describe the time-evolution of the probability of observing a certain number of point-like particles in each compartment at any given time. A typical example of a stochastic reaction–diffusion process is illustrated in Fig. 1a. In this scenario, each cell within a multi-cellular structure functions as a distinct compartment. The particles are molecules whose numbers inside each compartment fluctuate because of various chemical reactions confined to this compartment, as well as the exchange of molecules between this compartment and others through mechanisms such as pure diffusion [21] or active transport [22]. The primary challenge addressed in our study involves determining the joint or marginal distribution of molecular counts across all compartments, effectively capturing the spatial dynamics at the cellular level. Because the applications that we discuss later focus on biochemical reactions in living cells, from here onward we will use molecules instead of particles and voxels instead of compartments.

Whilst simulations using this approach are much faster than individual particle based methods, nevertheless they are still very computationally challenging when used to model realistic systems. For example, applying a conventional simulation technique, Finite State Projection (FSP [23]), to the RDME implies solving $N^{MS}$ coupled

* Corresponding author.
*E-mail addresses:* z.cao@queensu.ca (Z. Cao), ramon.grima@ed.ac.uk (R. Grima).

https://doi.org/10.1016/j.mbs.2024.109248
Received 20 January 2024; Received in revised form 7 May 2024; Accepted 3 July 2024
Available online 8 July 2024

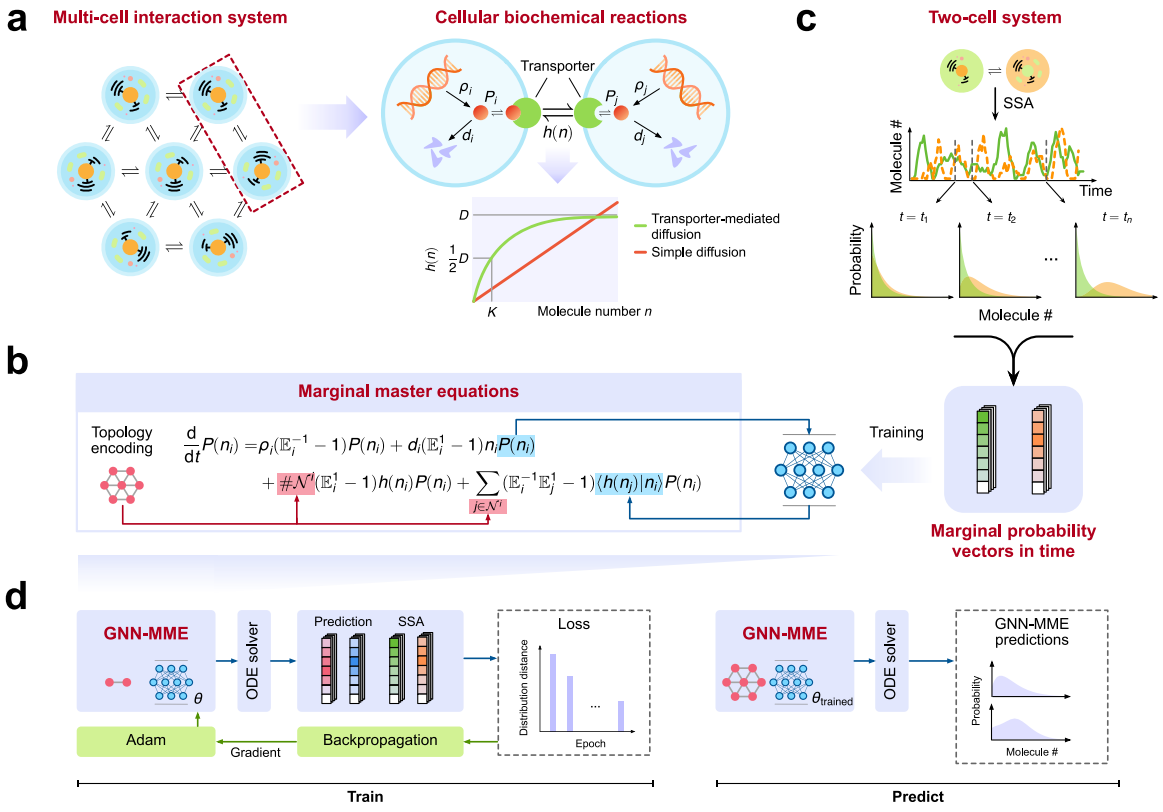



**Fig. 1.** Graph neural network aided construction of master equations for the marginal distributions of protein counts in a multi-cell system. (a) Illustration of a connected system of $m$ cells — proteins (solid red circles) are produced and degraded in cell $i$ with rates $\rho_i$ and $d_i$, respectively and they are transported between cells by membrane transport proteins (transporters) such as carrier proteins (green pac-mans). The propensity function modelling active transport $h(n)$ is a Michaelis–Menten function of the protein number which effectively captures the saturation of carrier proteins at high number. (b) From the master equation of the multi-cell system, a master equation for the marginal distributions is derived (Eq. (4)). The terms describing transport influx into a cell are unknown functions of the marginal distributions. Hence the master equations are not closed and cannot be solved. We use a neural network, trained using SSA simulations of two-cell systems (c), to learn the function that maps the marginal distributions to the effective propensity describing influx into a cell, thus closing the master equations — we call the latter a *graph neural network modularised master equation (GNN-MME)*. (d) Left panel. An optimisation procedure is used to find the neural network coefficients which minimise the distance between the FSP solution of the two-cell GNN-MME and the SSA marginal distributions. This constitutes the learning of the functional approximation of the effective influx propensity by the neural network. (d) Right panel. The learnt propensities from two-cell simulations are used to directly build a GNN-MME describing the full multi-cell system. The FSP of this master equation leads to predicted time-dependent marginal distributions of protein counts in all cells of an arbitrarily connected multi-cell network with arbitrary birth–death parameters.

differential equations (each equation describes the time-evolution of the probability of the system being in one of the allowable states) where $N-1$ is a user-defined upper bound on the molecule counts for all (chemical) species in each voxel, $M$ is the number of voxels and $S$ is the number of species. The use of Monte Carlo techniques such as Gillespie's stochastic simulation algorithm (SSA [24–28]) avoids the solution of these differential equations, but the computation time can still be very significant because (i) the effective number of species and the number of reactions increase linearly with $M$ and (ii) a considerably large amount of ensemble averaging of the trajectories may be needed to obtain statistically accurate results for the distributions of molecular counts in each voxel. This makes the practical exploration of a spatially extended system's properties across parameter space very challenging.

Recently a range of neural network based methods have been devised which significantly reduce the computational burden compared to conventional simulation methods [29–34]. A feedforward neural network was used in Ref. [29] to construct Markovian approximations of stochastic biochemical reaction systems with non-Markovian dynamics. DeepCME [30] uses reinforcement learning to estimate the moments of molecular counts from stochastic simulations. Mixture Density Networks have been used to learn the transition kernel of the chemical master equation [31]. Nessie (Neural Estimation of Stochastic Simulations for Inference and Exploration) [32] uses a neural-network approach to learn a negative binomial mixture approximation to the marginal distributions of each species. A similar approach to the latter but extending the prediction to steady-state joint distribution solutions for a two-species model of the life cycle of RNA was recently reported in Ref. [33]. More recently, variational autoregressive networks were used to solve the chemical master equation [34]. We emphasise that in all these cases, the systems of interest are non-spatial (well-mixed) chemical reaction systems. However, in principle, some of these methods could be adapted for spatially-extended systems. In practice this would be considerably difficult because the neural networks in these methods try to learn a mapping from the very large dimensional parameter space (which scales with the number of voxels) to the joint or marginal distributions (or their moments).

Inspired by these recent developments, in this article we devise a novel method which uses neural networks to circumvent the issues of conventional simulation methods (FSP and SSA). Specifically, we employ graph neural networks (GNNs) to improve the capability of a well-trained neural network to generalise predictions to unseen reaction networks. A GNN is a type of neural network structure operating on graph-structured data which has found several applications in RNA-seq data analyses [35,36], biomedicine [37] and material science [38]. It has powerful capabilities of extrapolation to different datasets and topologies, and avoids overfitting [39,40]. Here we utilise GNNs to learn, from a small sample of SSA simulations, the propensity functions of an effective master equation for the marginal distributions of each species in each voxel. Application of FSP to the latter implies the solution of $M$ (voxel number) $\times$ $N$ (maximal counts for any species $+1$) $\times$ $S$ (species number) equations, typically a very small fraction of the $N^{MS}$ equations given by the conventional FSP. In addition, the

method's ability to extrapolate predictions to voxel network topologies and regions of parameter space unseen in its training gives it a distinctive computational advantage over the SSA.

## 2. Results

### 2.1. Graph neural networks provide an accurate and computationally cheap approximation of the RDME: illustration on a single-species system

We illustrate our novel approach by considering a system of $m$ cells with biochemical reactions occurring inside them and the transport of molecules between them. Specifically in each cell $i$, we model the production of proteins by a zero-order reaction with rate $\rho_i$ and their degradation/dilution by a first-order reaction with rate $d_i$. Additionally, we model the active transport of protein molecules from cell $i$ to a neighbouring cell $j$ by a first-order reaction with a Michaelis–Menten propensity $h(n) = \frac{Dn}{K+n}$ where the saturation at large $n$ stems from the finite maximum speed of the transporter molecules. Note that the Michaelis–Menten propensity follows from a set of reactions with elementary propensities and well separated timescales [41,42]. This is an elementary model of a tissue [9]. The reaction scheme describing this system is given by:

$$\varnothing \xrightarrow{\rho_i} P_i, \qquad P_i \xrightarrow{d_i} \varnothing, \qquad P_i \underset{h(n_j)}{\overset{h(n_i)}{\rightleftharpoons}} P_j \tag{1}$$

for $i = 1, \dots, m$ and $j \in \mathcal{N}^i$ where $\mathcal{N}^i$ is the set of neighbours of cell $i$ which effectively encodes the topology of cell–cell interactions. The multi-cell system and the associated reactions are also illustrated in Fig. 1a. If the reaction–diffusion process was occurring in continuous space, in the limit of large number of molecules, the dynamics would be described by a PDE

$$\frac{\partial}{\partial t} p(x, y) = \Delta h(p(x, y)) + \rho(x, y) - d(x, y)p(x, y), \tag{2}$$

where $p(x, y)$ is the concentration of proteins ($P$) at spatial coordinate $(x, y)$, the production rate $\rho$ and degradation rate $d$ are position dependent, and $\Delta$ is Laplacian operator. However since we are interested in a description that is valid for all molecule numbers and there is a natural lattice (the cells) for the biological application under study, we instead utilise the master equation approach.

Specifically assuming Markovian dynamics, the stochastic dynamics are described by the RDME:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} P(n_1, \dots, n_m, t) = &\sum_{i=1}^{m} \rho_i(\mathbb{E}_i^{-1} - 1)P(n_1, \dots, n_m, t) + d_i(\mathbb{E}_i^{1} - 1)n_i P(n_1, \dots, n_m, t) \\ &+ \sum_{i=1}^{m} \sum_{j \in \mathcal{N}^i} \Big[(\mathbb{E}_i^{1}\mathbb{E}_j^{-1} - 1)h(n_i)P(n_1, \dots, n_m, t) \\ &+ (\mathbb{E}_i^{-1}\mathbb{E}_j^{1} - 1)h(n_j)P(n_1, \dots, n_m, t)\Big],\end{aligned} \tag{3}$$

where $P(n_1, \dots, n_m, t)$ is the probability of observing at time $t$, cell 1 with $n_1$ protein molecules, cell 2 with $n_2$ protein molecules, …, cell $m$ with $n_m$ protein molecules. The operator $\mathbb{E}_i^j$ is a step operator defined as $\mathbb{E}_i^j f(n_1, \dots, n_i, \dots, n_m) = f(n_1, \dots, n_i + j, \dots, n_m)$. If we wanted to solve Eq. (3) using the FSP algorithm, on a state space truncated up to a maximum molecule number $N - 1$, then we would need to integrate $N^m$ coupled differential equations. It is clear that for very modest maximum protein numbers, say a few tens of molecules, this algorithm is computationally prohibitive if we are interested in studying a system with more than a handful of interconnected cells.

To overcome this difficulty, we proceed in a different manner. First we use Eq. (3) to derive an exact master equation describing the time-evolution of the marginal distribution $P(n_i, t)$ of protein numbers in cell $i$ (see Methods)

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} P(n_i) =&\rho_i[P(n_i - 1) - P(n_i)] + d_i[(n_i + 1)P(n_i + 1) - n_i P(n_i)] \\ &+ \#\mathcal{N}^i \left[h(n_i + 1)P(n_i + 1) - h(n_i)P(n_i)\right] \\ &+ \sum_{j \in \mathcal{N}^i} [\langle h(n_j)|n_i - 1\rangle P(n_i - 1) - \langle h(n_j)|n_i\rangle P(n_i)],\end{aligned} \tag{4}$$

where time $t$ is suppressed for convenience, $\#\mathcal{N}^i$ is the number of neighbours of cell $i$ and $\langle h(n_j)|n_i\rangle = \sum_{n_j} h(n_j)P(n_j|n_i)$ is the effective transport propensity describing the influx of molecules from cell $j$ to cell $i$. Since generally the explicit dependence of the latter on $n_i$ is unknown, it follows that an exact solution of Eq. (4) is impossible. Instead we hypothesise that a neural network's ability to perform universal function approximation [43] may be conveniently used to establish a mapping from $P(n_i)$ and $P(n_j)$ to the effective transport propensity $\langle h(n_j)|n_i\rangle$. This implies that Eq. (4) is now transformed into a set of closed equations for $P(n_i)$. It then follows that if we apply the FSP algorithm on a truncated state space, assuming a maximum molecule number $N-1$, we would need to solve $Nm$ coupled differential equations. This is a vast improvement over the $N^m$ equations needed to solve the original stochastic description. This solution is acceptable provided of course we are not interested in the joint distribution function but *only* in the marginals.

Formalising this reasoning, it follows that Eq. (4) can be approximated by a new master equation of the form:

$$\frac{\mathrm{d}\mathbf{P}_i}{\mathrm{d}t} = \left[\rho_i \mathbf{A} + d_i \mathbf{B} + \#\mathcal{N}^i \mathbf{C}_h + \sum_{j \in \mathcal{N}^i} \mathbf{N}_\theta^{j \to i}\right] \mathbf{P}_i, \tag{5}$$

where the vector $\mathbf{P}_i$ collects all the probability elements of cell $i$, i.e., $\mathbf{P}_i = [P_i(0), P_i(1), \dots, P_i(N-1)]^\top \in \mathbb{R}^N$ and the matrices are defined as

$$\mathbf{A} = \mathrm{Span}_\downarrow(\mathbf{1}), \quad \mathbf{B} = \mathrm{Span}_\uparrow(\mathbf{I}), \quad \mathbf{C}_h = \mathrm{Span}_\uparrow(\mathbf{h}), \quad \mathbf{N}_\theta^{j \to i} = \mathrm{Span}_\downarrow(\mathbf{NN}_\theta^{j \to i}) \tag{6}$$

with $\mathbf{1} = [1, \dots, 1]^\top \in \mathbb{R}^{N-1}$, $\mathbf{I} = [1, 2, \dots, N-1]^\top \in \mathbb{R}^{N-1}$, $\mathbf{h} = [h(1), h(2), \dots, h(N-1)]^\top \in \mathbb{R}^{N-1}$ and $\mathbf{NN}_\theta^{j \to i} = [\mathrm{NN}_\theta^{j \to i}(1), \dots, \mathrm{NN}_\theta^{j \to i}(N-1)]^\top \in \mathbb{R}^{N-1}$ are the neural network outputs. The definitions of the two operators $\mathrm{Span}_\uparrow(\cdot)$ and $\mathrm{Span}_\downarrow(\cdot)$, both of which map a vector $\mathbf{a} = [a_1, \dots, a_{N-1}]^\top$ into sparse matrices, are given by

$$\mathrm{Span}_\uparrow(\mathbf{a}) = \begin{bmatrix} 0 & a_1 & 0 & \cdots & 0 & 0 \\ 0 & -a_1 & a_2 & \cdots & 0 & 0 \\ 0 & 0 & -a_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -a_{N-2} & a_{N-1} \\ 0 & 0 & 0 & \cdots & 0 & -a_{N-1} \end{bmatrix}_{N \times N},$$

$$\mathrm{Span}_\downarrow(\mathbf{a}) = \begin{bmatrix} -a_1 & 0 & 0 & \cdots & 0 & 0 \\ a_1 & -a_2 & 0 & \cdots & 0 & 0 \\ 0 & a_2 & -a_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -a_{N-1} & 0 \\ 0 & 0 & 0 & \cdots & a_{N-1} & 0 \end{bmatrix}_{N \times N}$$

Specifically, the neural network $f_\theta$ parametrised by neural network coefficients $\theta$ (the weights and biases) establishes the mapping

$$f_\theta : (\mathbf{P}_i, \mathbf{P}_j) \to \mathbf{NN}_\theta^{j \to i},$$

which is independent of the birth–death kinetic parameters $\rho_i$ and $d_i$. We shall refer to Eq. (5) as the *graph neural network modularised master equation (GNN-MME)*. See Fig. S1 for an illustration and interpretation of Eq. (5) in the framework of GNNs. We also note that our approach falls under the umbrella of Universal Differential Equations [44] and it is similar to a special type of GNN called a graph network [45].

Finally we devise an effective neural network training procedure. To keep the computational cost to a minimum, we propose to learn the neural network coefficients $\theta$ of the multi-cell system using SSA simulations of a system with only two neighbouring cells for different sets of birth–death kinetic parameters $\rho_i$ and $d_i$ but with constant transport parameters $D$ and $K$ (Figs. 1b, c). Note that we fixed the latter two parameters to reflect the underlying assumption of our model that the diffusion mechanism is independent of the topology of cell–cell interactions and all other reaction parameter values. This approach forces the neural network to learn the dynamic behaviour of the system under a consistent set of physical conditions, thereby ensuring the generalisability of the model. In Fig. 1b, the topology encoding stands for an adjacency matrix between adjacent cells/voxels. From these simulations we obtain the time-dependent marginal distributions of protein counts in the two cells, $\mathbf{P}_1^{\mathrm{SSA}}$ and $\mathbf{P}_2^{\mathrm{SSA}}$. Subsequently, the optimal neural network coefficients are found by minimising a loss function between the latter and the time-dependent distributions given by the FSP solution of the GNN-MME for a two-cell system (Fig. 1d left panel), specifically,

$$J(\theta) = \sum_{i=1,2} \sum_{j} \mathrm{MSE}(\mathbf{P}_i^{\mathrm{SSA}}(t_j), \mathbf{P}_i(t_j|\theta))$$

where MSE stands for mean-squared error and $t_j$ is the sampling time. This completes the learning of the effective propensity. By substituting the latter in a GNN-MME for a multi-cell network we can predict the stochastic dynamics for any arbitrarily connected network (Fig. 1d right panel).

The reliability of this training procedure is tested next. Using three parameter sets (Table S1) we verified that once the neural network was well trained, the FSP solution of the GNN-MME gave time-dependent marginal distributions of protein counts in the two cells that were practically indistinguishable from those obtained using the SSA (Fig. 2a). Subsequently we tested if given the learnt effective propensity (from a two-cell system) we could directly build a GNN-MME to predict the stochastic dynamics of an arbitrarily connected multi-cell network and using birth–death parameters different than the three training sets but with the same transport parameters (Table S2). In Fig. 2c, Figs. S2 and S3 we show the predictions of the GNN-MME for networks of cells connected together in the shape of five letters (Fig. 2b). In all cases, we found that the marginal distribution of protein counts for any cell, at any point in time, agrees very well with distributions obtained from SSA simulations of the multi-cell networks.

We emphasise that the complexity of the RDME, which resists exact solutions through standard analytical methods, underlines the novelty and significance of our neural network's success. This complexity means that the network's accurate predictions cannot be ascribed to a simple representation of the joint distribution $P(n_1, n_2, \ldots, n_m)$. In simpler scenarios where the transport propensity $h(n)$ is linear, a trivial steady-state solution exists in the form of a product Poisson joint distribution. However, our case involves a non-linear $h(n)$, as evidenced by the Fano factor (variance divided by the mean) of protein counts in each cell exceeding 1 in steady-state conditions (see Figs. S2 and S3). This non-linearity adds to the RDME's complexity, further highlighting the advanced capability of our GNN-based approach in capturing the intricate dynamics of the system.

We have shown that the GNN-MME accurately approximates the marginal protein distribution solutions of the RDME Eq. (4). The remaining question is why our neural-network based approach would be a useful strategy compared to conventional numeric or simulation methods. Earlier we discussed the vast improvement compared to a direct application of the FSP to solve Eq. (4). Now we seek to understand the computational advantage that the GNN-MME approach affords over the direct computation of the distributions of a multi-cell cellular system from a large number of SSA samples. In Figs. 2d–f (top panel), we compare the two methods' accuracy of the marginal distributions and the computation time for a system of 10 cells connected as a ring and each having the same parameters. The two methods provide comparable accuracy (relative to the "true" solution) given the two-cell SSA simulations used for training the GNN-MME have the same sample size as the 10-cell SSA simulations. However the use of the GNN-MME approach leads to computational savings. Here we note that in our estimation of the computation time for the GNN-MME we did not factor in the training time for neural networks. While this may be important in other studies, here it is not particularly relevant because once the neural network is trained using two-cell SSA simulations, it can cast predictions on an infinite number of larger scale systems and hence effectively the training time for any given large scale system (such as the 10-cell system in this example) is negligible. In Figs. 2d–f (bottom panel), we repeat the comparison of the accuracy and the computation time but now for the more challenging case where the parameters of the GNN-MME are different than those used for the SSA training of the two-cell system. Use of the GNN-MME approach leads to an impressive $\sim 300$ times decrease in the computation time compared to the SSA of the full system while maintaining similar accuracy.

## 3. The neural network based approach is extensible to multi-species systems: spatio-temporal mRNA & protein patterns at subcellular resolution

Thus far we have illustrated the GNN based approach using a single-species system. We next consider its generalisation to a multi-species system which will require the use of additional neural networks. To be specific, we consider a spatially extended version of a simple "well-mixed" model of a genetic negative feedback loop [46,47]. We divide the two-dimensional cross-section of a cell into an $11 \times 11$ voxel grid; a voxel represents either a small volume around the gene locus, or a small spatial domain in the nucleus or cytoplasm. This grid size allows for a comprehensive demonstration of how our method manages increased complexity and spatial extent, which is crucial for applications that require fine spatial resolution. We define a set of reactions to model the following well-known subcellular phenomena [48]. Messenger RNA (mRNA) is produced at the gene locus with a rate that is inversely proportional to the number of proteins in the vicinity of the locus, mRNA diffuses in the nucleus and can be transported into the cytoplasm where it is translated into protein and it can also degrade. Proteins diffuse in the cytoplasm and can also be transported to the nucleus where they can diffuse to the gene locus and affect the mRNA production rate. Note that by diffusion here we mean simple diffusion, not the active transport we considered earlier. This system of reactions constitutes a negative feedback loop since an increase in the mRNA levels leads to a similar increase in protein levels which after a time lag leads to a decrease in the mRNA levels. The fine-grained cell and the associated reactions are illustrated in Fig. 3a.

If we use the same GNN approach as in the previous single-species example, the resulting GNN-MME is a set of equations for the joint probability distribution of mRNA and protein numbers in each voxel. This implies we need to solve $121N^2$ differential equations simultaneously, which while it is a significant improvement over the $N^{242}$ differential equations solved using the standard FSP, nevertheless it is still quite a large number if $N$ is larger than a handful of molecules. To further reduce the computation time, what we need are differential equations for the marginal distributions of each of the two species in each voxel. To achieve this aim, we first write the exact master equation for the marginal distributions of mRNA and protein in each voxel and then construct a GNN-MME by means of two neural networks: (i) an intervoxel reaction neural network is used to find an approximate mapping from the marginal distributions of mRNA or protein in the two adjacent voxels to the effective influx diffusion propensity; (ii) an intravoxel reaction neural network is used to find an approximate mapping of the marginal distributions of mRNA and protein in the same voxel to the effective transcription and translation propensities. The construction of the GNN-MME is illustrated in Fig. 3b. Specifically, for

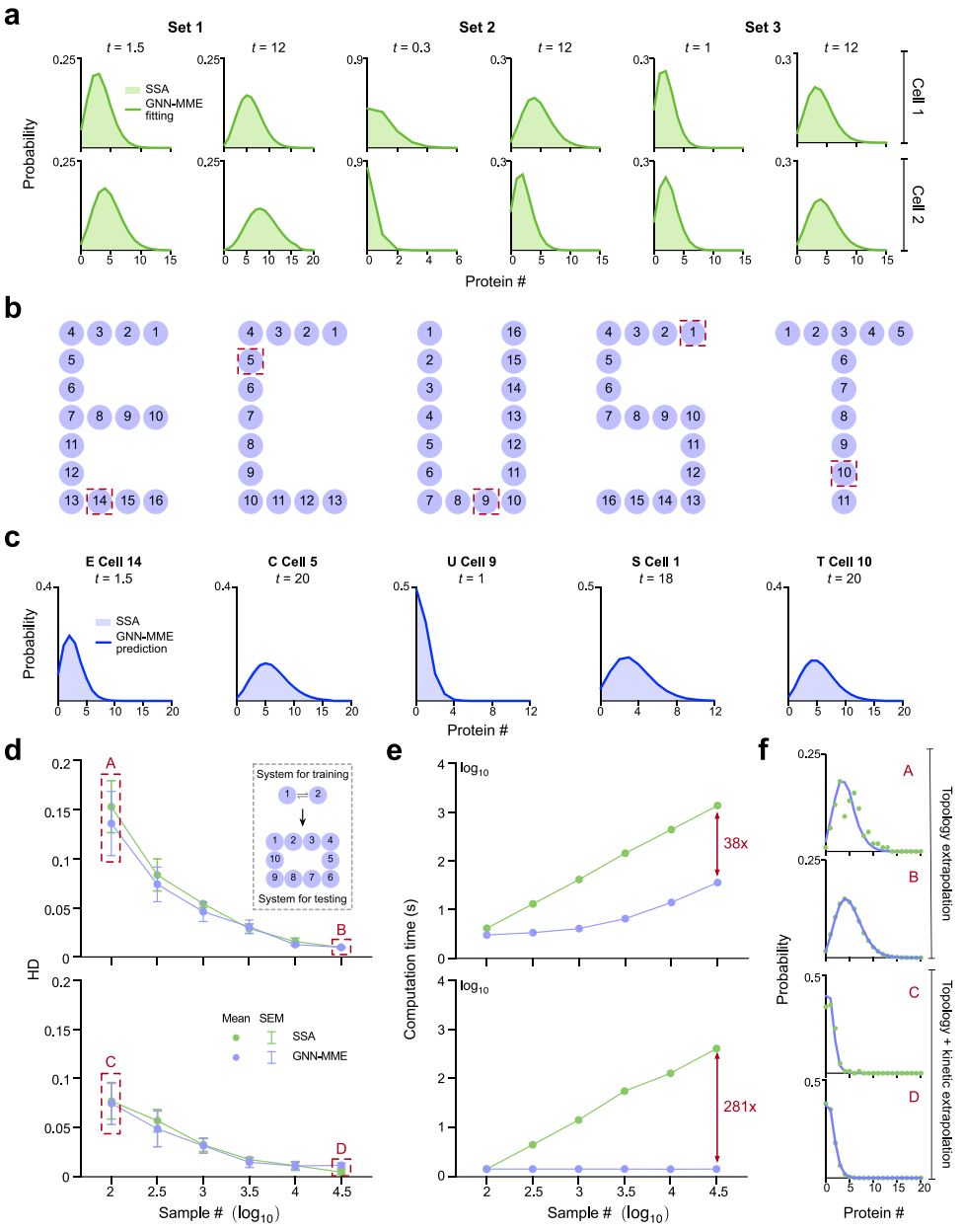



**Fig. 2.** Evaluation of the performance of the GNN-MME for the system of connected cells shown in Fig. 1a. (a) The neural network was trained on two-cell SSA simulations with three different sets of kinetic parameters (Table S1). The optimisation algorithm finds parameters of the neural network that leads to an excellent agreement between the distributions predicted by the two-cell GNN-MME and the SSA distributions used for training. The fitting error is summarised in Table S5. (b, c) The multi-cell GNN-MME with neural networks trained from the 3 parameter sets of two-cell systems accurately predicts the protein distributions for cells in 5 different letter-like topologies where the birth–death kinetic parameters were chosen from uniform distributions $\rho_i \sim U_{[0,5]}$ and $d_i \sim U_{[0,1]}$ while the transport parameters were fixed to $D = 10$ and $K = 5$ (Table S2). The corresponding prediction error is reported in Table S6. (d–f) (Top panels) We furthermore tested the accuracy and computational performance of the GNN-MME using a circular system of 10 identical cells with $\rho_i = 2.5$, $d_i = 0.5$ $\forall i$, $D = 10$ and $K = 5$. The GNN-MME trained with a specified number of SSA samples of a two-cell system gave steady-state marginal distributions of comparable accuracy to those directly obtained using the same number of SSA samples of the 10-cell system — the accuracy was measured by the Hellinger distance (HD) between the distributions and the ground truth (a 10-cell SSA simulation with $10^7$ samples). However the computation time of the GNN-MME (time to compute the specified SSA samples for the two-cell system plus the time for the GNN-MME to make predictions) is typically considerably less than that of the same number of 10-cell SSA simulations. (d–f) (Lower panels) Same as for the top panels but where the system for testing had different parameters ($\rho_i = d_i = 1$ $\forall i$) unused for training. Note that in this case the computation time is purely the GNN-MME prediction time since SSA data was not collected. For a fair comparison, benchmarking was performed on one core of an Intel Xeon Gold 6246R 3.40GHz× 64-core CPU 376.6G RAM. See Methods for technical specifications of the neural networks.

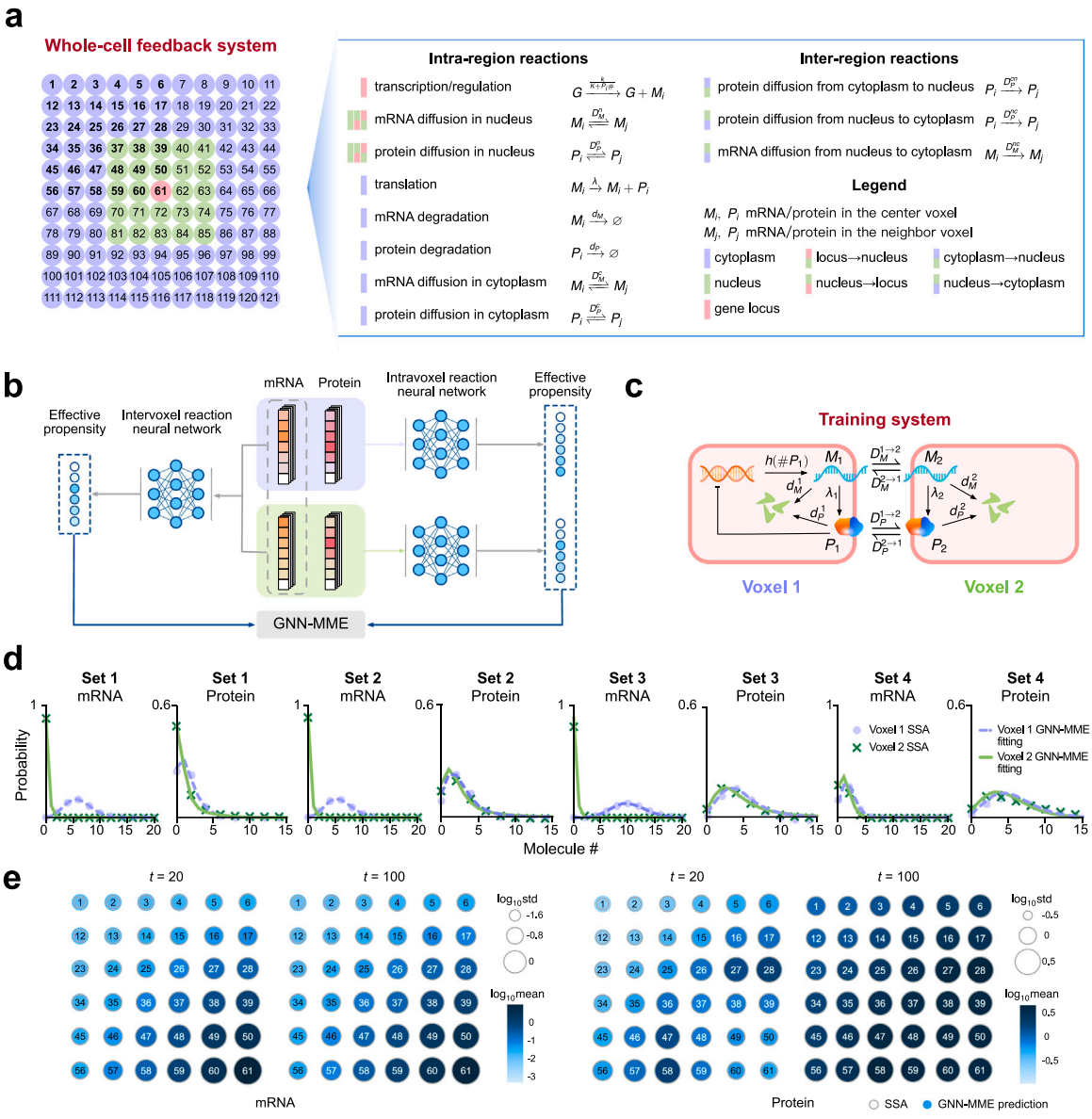



**Fig. 3.** GNN-MME approach enables accurate prediction at sub-cellular resolution of mRNA and protein dynamics across a whole cell from stochastic simulations of a two-voxel system. (a) The 2D cross section of a cell is divided into 11 × 11 voxels; the central 5 × 5 voxels represent the nucleus (green), the centre represents a small nuclear region containing the gene locus (red), and the rest represent the cytoplasm (purple). Reactions are defined describing transcriptional regulation via protein binding, translation, diffusion and degradation. mRNA degradation can only occur in the cytoplasm and while protein can move in and out of the nucleus, mRNA movement is only allowed from the nucleus to the cytoplasm. (b) We use neural networks to learn an approximate mapping (i) from the marginal distributions of mRNA or protein in two adjacent voxels (shown in blue and green) to the effective influx diffusion propensity (only one of these two intervoxel reaction neural networks is shown) and (ii) from the marginal distributions of mRNA and protein in the same voxel to the effective transcription and translation propensities. These propensities enable us to construct a GNN-MME (Eqs. (7) and (8)) whose solution using FSP directly gives approximate marginal distributions of mRNA and protein. (c) Training of the neural networks is performed using a two-voxel system which while much simpler than the whole cell system, it describes the same phenomena but on a much smaller spatial scale. (d) SSA simulations of the two-voxel system with four sets of parameters (Table S3) were used to train the intra- and intervoxel neural networks. The optimisation algorithm finds parameters of the neural networks that lead to excellent agreement between the distributions predicted by the two-voxel GNN-MME (Eq. (11)) and the SSA distributions used for training. (e) The propensities learnt from the two-voxel simulations are used to construct the GNN-MME of the whole cell system in (a) (parameters in Table S4). Its FSP solution leads to time-dependent protein and mRNA distributions whose first and second moments are in good agreement with SSA ($10^5$ samples) of the whole cell system — the colour and size of the shaded circles show the logarithmic mean and the logarithmic standard deviation (std) predicted by the GNN-MME, respectively; the open grey circles show the logarithmic std predicted by SSA. For a comparison of the distributions see Figs. S4–S6. See Methods for technical specifications of the neural networks.

voxels in the nucleus (including the gene locus), the GNN-MME takes the form

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{P}_{M_i} = \Big[\underbrace{k\mathbf{N}^i_{\mathrm{intra}_M}\delta_g}_{\text{part A}} + \underbrace{D^n_M \sum_{j\in\mathcal{N}^i_n}\mathbf{N}^{j\to i}_{\mathrm{inter}_M}}_{\text{part B}} + \underbrace{\left(\#\mathcal{N}^i_n D^n_M + \#\mathcal{N}^i_c D^{nc}_M\right)\mathbf{B}}_{\text{part C}}\Big]\mathbf{P}_{M_i}, \\ \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{P}_{P_i} = \Big[\underbrace{D^n_P \sum_{j\in\mathcal{N}^i_n}\mathbf{N}^{j\to i}_{\mathrm{inter}_P}}_{\text{part D}} + \underbrace{D^{cn}_P \sum_{j\in\mathcal{N}^i_c}\mathbf{N}^{j\to i}_{\mathrm{inter}_P}}_{\text{part E}} + \underbrace{\left(\#\mathcal{N}^i_n D^n_P + \#\mathcal{N}^i_c D^{nc}_P\right)\mathbf{B}}_{\text{part F}}\Big]\mathbf{P}_{P_i}, \end{cases} \tag{7}$$

while in the cytoplasm it has the form

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{P}_{M_i} = \Big[\underbrace{D^c_M \sum_{j\in\mathcal{N}^i_c}\mathbf{N}^{j\to i}_{\mathrm{inter}_M}}_{\text{part G}} + \underbrace{D^{nc}_M \sum_{j\in\mathcal{N}^i_n}\mathbf{N}^{j\to i}_{\mathrm{inter}_M}}_{\text{part H}} + \underbrace{\left(\#\mathcal{N}^i_c D^c_M + d_M\right)\mathbf{B}}_{\text{part I}}\Big]\mathbf{P}_{M_i}, \\ \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{P}_{P_i} = \Big[\underbrace{\lambda\mathbf{N}^i_{\mathrm{intra}_P}}_{\text{part J}} + \underbrace{D^c_P \sum_{j\in\mathcal{N}^i_c}\mathbf{N}^{j\to i}_{\mathrm{inter}_P}}_{\text{part K}} + \underbrace{D^{nc}_P \sum_{j\in\mathcal{N}^i_n}\mathbf{N}^{j\to i}_{\mathrm{inter}_P}}_{\text{part L}} \\ \qquad + \underbrace{\left(\#\mathcal{N}^i_c D^c_P + \#\mathcal{N}^i_n D^{cn}_P + d_P\right)\mathbf{B}}_{\text{part M}}\Big]\mathbf{P}_{P_i}. \end{cases} \tag{8}$$

Here $\mathbf{P}_{M_i}$ and $\mathbf{P}_{P_i}$ denote the marginal distributions of mRNA and protein counts in voxel $i$, respectively. The reactions associated with the various rate parameters in these equations can be found in Fig. 3a. The symbols $\mathcal{N}^i_n$ and $\mathcal{N}^i_c$ mean the set of neighbours in the nucleus and cytoplasm for voxel $i$ respectively. In Eq. (7), part A uses an intravoxel neural network to model the effect of protein-modulated transcription at the gene locus voxel ($\delta_g$ is a delta function and equal to 1 if and only if voxel $i$ is the gene locus), parts B and D use two intervoxel neural networks to describe the net diffusive influx of mRNA and protein into a nuclear voxel $i$ respectively, parts C and F describe the net diffusive outflux of mRNA and protein from a nuclear voxel $i$ respectively, and part E uses an intercellular neural network to describe the diffusive influx of protein from adjacent cytoplasmic voxels to the nuclear voxel $i$. In Eq. (8), parts G and H describe the diffusive influx of mRNA from adjacent cytoplasmic and nuclear voxels to cytoplasmic voxel $i$ respectively, the same for parts K and L but for protein, part J uses an intravoxel network to describe translation, part I describes diffusive outflux and degradation of mRNA in a cytoplasmic voxel, and part M describes the same effects but for protein.

The matrix $\mathbf{B}$ is the same as defined in Eq. (6). The four neural network matrices in Eqs. (7) and (8) are $\mathbf{N}^i_{\mathrm{intra}_M} = \mathrm{Span}_{\downarrow}(\mathbf{NN}^i_{\mathrm{intra}_M})$, $\mathbf{N}^i_{\mathrm{intra}_P} = \mathrm{Span}_{\downarrow}(\mathbf{NN}^i_{\mathrm{intra}_P})$, $\mathbf{N}^{j\to i}_{\mathrm{inter}_M} = \mathrm{Span}_{\downarrow}(\mathbf{NN}^{j\to i}_{\mathrm{inter}_M})$ and $\mathbf{N}^{j\to i}_{\mathrm{inter}_P} = \mathrm{Span}_{\downarrow}(\mathbf{NN}^{j\to i}_{\mathrm{inter}_P})$. The associated four neural networks learn the following mappings

$$\begin{aligned} &f_{\theta_1,\mathrm{intra}_M} : (\mathbf{P}_{M_i}, \mathbf{P}_{P_i}) \to \mathbf{NN}^i_{\mathrm{intra}_M}, \\ &f_{\theta_2,\mathrm{intra}_P} : (\mathbf{P}_{M_i}, \mathbf{P}_{P_i}) \to \mathbf{NN}^i_{\mathrm{intra}_P}, \\ &f_{\theta_3,\mathrm{inter}_M} : (\mathbf{P}_{M_i}, \mathbf{P}_{M_j}) \to \mathbf{NN}^{j\to i}_{\mathrm{inter}_M}, \\ &f_{\theta_4,\mathrm{inter}_P} : (\mathbf{P}_{P_i}, \mathbf{P}_{P_j}) \to \mathbf{NN}^{j\to i}_{\mathrm{inter}_P}. \end{aligned} \tag{9}$$

We emphasise that given Eqs. (7) and (8), the number of differential equations that need to be solved is reduced from $N^{242}$ (standard FSP) to just $242N$.

What remains is to define a training protocol so that we can efficiently learn the optimal neural network coefficients $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]^{\mathsf{T}}$ from stochastic simulations of a system that is much smaller in scale than the whole cell system but which effectively models the same phenomena. Thus for training purposes, we consider a two-voxel, coarse-grained version of the whole cell model with the following set of reactions (also illustrated by a cartoon in Fig. 3c).

$$\begin{aligned} &G \xrightarrow{h(n)} G + M_1, \quad M_1 \xrightarrow{\lambda_1} M_1 + P_1, \quad M_1 \xrightarrow{d^1_M} \varnothing, \quad P_1 \xrightarrow{d^1_P} \varnothing, \\ &M_2 \xrightarrow{\lambda_2} M_2 + P_2, \quad M_2 \xrightarrow{d^2_M} \varnothing, \quad P_2 \xrightarrow{d^2_P} \varnothing, \quad M_1 \underset{D^{2\to1}_M}{\overset{D^{1\to2}_M}{\rightleftharpoons}} M_2, \quad P_1 \underset{D^{2\to1}_P}{\overset{D^{1\to2}_P}{\rightleftharpoons}} P_2, \end{aligned} \tag{10}$$

where $h(n) = \frac{k}{K+n}$. To train the four neural networks in (9), we need to derive the GNN-MME for the two-voxel system, which is indeed a reduced version of Eqs. (7) and (8):

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{P}_{M_i} = \left[k\mathbf{N}^i_{\mathrm{intra}_M}\delta_1 + D^{j\to i}_M\mathbf{N}^{j\to i}_{\mathrm{inter}_M} + \left(D^{i\to j}_M + d^i_M\right)\mathbf{B}\right]\mathbf{P}_{M_i}, \\ \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{P}_{P_i} = \left[\lambda_i\mathbf{N}^i_{\mathrm{intra}_P} + D^{j\to i}_P\mathbf{N}^{j\to i}_{\mathrm{inter}_P} + \left(D^{i\to j}_P + d^i_P\right)\mathbf{B}\right]\mathbf{P}_{P_i}, \end{cases} \tag{11}$$

for $i = 1, 2$ and $j \in \mathcal{N}^i$. The delta function $\delta_1 = 1$ if and only if $i = 1$. The training dataset consists of the SSA marginal distributions of mRNA and protein for both voxels for four different sets of kinetic parameters (Table S3). In Fig. 3d we confirm that the neural networks are well trained by an excellent agreement between the GNN-MME for the two-voxel system and the SSA distributions used for training.

Using the learnt propensities from the two-voxel system, we then build the GNN-MME for the whole cell 121-voxel system shown in Fig. 3a, i.e. Eqs. (7)–(8). Note that some of the parameters of the whole cell system (Table S4) are different than those used for training the two-voxel system. In Fig. 3e we show the mean and variance of mRNA and protein counts for a quadrant of the whole cell system (the other quadrants show the same information because of the spatial symmetry of the system) at two different times; for the corresponding marginal distributions see Figs. S4–S6. The excellent agreement with the distributions computed using the SSA of the whole cell system confirms the GNN-MME's ability to accurately predict the fine-grained spatio-temporal pattern of mRNA and protein dynamics at sub-cellular resolution from SSA stochastic trajectories of a coarse-grained, two-voxel model. The results also identify an interesting dependence of the Fano factor of protein fluctuations with distance from the gene locus, e.g. in Fig. S5 from voxel 56 to 61, the Fano factor increases as we move from the cell boundary to the nucleus–cytoplasm boundary then decreases as we move into the nucleus and finally again increases as it approaches the gene locus.

To assess the robustness of our method against variations in initial conditions, we applied neural networks, already well-trained on two-voxel systems, to predict the temporal evolution of mRNA and protein marginal distributions in the aforementioned $11 \times 11$-voxel system. We initiated this process by setting the mRNA levels in voxel 61 according to Poisson distributions with mean values of 5 and 10. Remarkably, we observed excellent consistency in the marginal distributions of both mRNA and proteins across all voxels, as detailed in Figures S7 to S10.

## 4. Discussion

In this article, we have described a novel method which uses GNNs to automatically "derive" a master equation describing the time-evolution of the marginal probability distributions of gene products in each voxel of a spatially extended system. The numerical solution of this effective master equation leads to solutions for the marginal distributions in a small fraction of the time of conventional methods, thereby potentially enabling the large scale simulation of stochastic gene expression at the sub-cellular resolution for every cell in a tissue.

While it is possible to write a master equation for the marginal distributions from first principles, this equation cannot be generally solved because it is not closed due to the presence of terms which are not functions of the marginal distributions. We circumvent this issue by using a neural network to learn (from a small batch of stochastic simulations) these terms as functions of the marginal distributions,

thereby closing the master equation and enabling its solution using FSP. Whilst we showcased the procedure using one- and two-species examples, it is general for a system with any number of voxels and species. It is straightforward to deduce that the direct solution of this effective master equation for a spatially extended system with $M$ voxels and $S$ species, assuming an upper bound $N-1$ for the molecule counts in each voxel, implies the solution of $N \times M \times S$ coupled ordinary differential equations. By contrast, solving the reaction–diffusion master equation describing the full spatial stochastic dynamics using FSP would imply solving $N^{MS}$ ordinary differential equations. To put this in perspective, if we want to model the spatial variation of transcript numbers of one gene inside a single cell which has been discretised into 100 voxels, with each voxel holding at most 1 transcript, the conventional FSP requires the solution of $10^{30}$ coupled differential equations while our method reduces the computation to the solution of 200 coupled differential equations! We also showed that the method's computation time is often much smaller than Monte Carlo methods such as the SSA — fundamentally this is because the neural network only needs two-voxel simulations to train it and then it can extrapolate the predictions to an arbitrarily connected voxel network with different kinetic parameter values. The method is promising for whole cell modelling since we showed that neural network training on a coarse-grained, two-voxel version of a cell is sufficient to cast accurate predictions for much finer discretisations of intracellular space.

A limitation of our method is that it cannot give the joint distributions of molecular counts in all voxels. The method can either give the marginal distributions of each species in each voxel or else if one uses the neural network to only approximate the transport influx terms then the method gives the joint distributions of all species in each voxel. We do not see this as a major hindrance since often this is all the information that is needed. We note that while it might be possible for some systems to analytically approximate the terms in the master equations for the marginals [49–51] which the GNN targets, such a procedure would be based on some implicit assumption (akin to moment-closure techniques) and hence its accuracy would generally be less compared to the automated technique we have described.

We found in practice, the prediction accuracy of GNN-MME decreases as the diffusion coefficient increases. This is potentially because Eq. (5) implicitly encodes the assumption that GNN-MME only collects information from the first-order neighbours (see Eq. (5)), while the effect of high-order neighbours becomes increasingly important when the diffusion coefficient is large. This is also known as the oversmoothing issue of GNN. A remedy would involve introducing a virtual super voxel that exchanges information with all other voxels.

We note that recently another approach, one based on mixture density networks, was used to approximate the transition kernel of the RDME and thus accelerate the simulation of spatial stochastic reaction systems, such as that modelling honeybee colony defense [52]. Our approach is different because the neural network is specifically designed to capture dynamic characteristics associated with diffusion, thus enabling application to reaction networks across various cell–cell interaction topologies. Moreover, the approach in Ref. [52] approximates the transition kernel for a fixed time step, hence restricting its predictions to fixed time grids. By contrast, our method integrates the neural network within the marginal master equations, formulated as ordinary differential equations (ODEs), allowing for predictions at any desired temporal resolution.

Concluding, our approach overcomes existing challenges by training a neural network on two-voxel simulations to effectively derive a master equation for the marginal distributions in each voxel within a large, complex spatially-extended system. This novel capability for topological extrapolation suggests that our method could facilitate large-scale simulations of spatially extended systems, which are currently unachievable with conventional techniques. For instance, our method could significantly enhance the modelling of spatio-temporal dynamics of the SIR model on a mesh-grid. Specifically, by training neural networks on historical epidemic data from small regions, our approach could potentially accurately predict the progression of a novel outbreak on a national scale. This and other applications are currently under investigation.

## 5. Methods

### 5.1. Derivation of the marginal master equation Eq. (4)

By taking the sum of $n_k$, $k \neq i$ on both sides of Eq. (3), we obtain

$$\sum_{n_k} \frac{\mathrm{d}}{\mathrm{d}t} P(n_1, \dots, n_m) = \frac{\mathrm{d}}{\mathrm{d}t} \sum_{n_k} P(n_1, \dots, n_k, \dots, n_m) = \frac{\mathrm{d}}{\mathrm{d}t} P(n_1, \dots, n_{k-1}, n_{k+1}, \dots, n_m),$$

where the last step follows from the law of total probability. By sequentially repeating the above procedure for all $n_k$, $k \neq i$, the term becomes

$$\sum_{n_k, \forall k \neq i} \frac{\mathrm{d}}{\mathrm{d}t} P(n_1, \dots, n_m) = \frac{\mathrm{d}}{\mathrm{d}t} P(n_i). \tag{12}$$

For the first term on the right hand side of Eq. (3), taking the sum over $n_k$ and using the same procedure, we obtain

$$\begin{aligned}\sum_{n_k, \forall k \neq i} \rho_j (\mathbb{E}_j^{-1} - 1) P(n_1, \dots, n_m) =& \rho_j \sum_{n_k, \forall k \neq i} \big[P(n_1, \dots, n_j - 1, \dots, n_m) \\ & - P(n_1, \dots, n_j, \dots, n_m)\big] \\ =& \rho_j \sum_{n_j} [P(n_i, n_j - 1) - P(n_i, n_j)] \\ =& \rho_j [P(n_i) - P(n_i)] = 0\end{aligned}$$

for $j \neq i$, and

$$\begin{aligned}\sum_{n_k, \forall k \neq i} \rho_i (\mathbb{E}_i^{-1} - 1) P(n_1, \dots, n_m) =& \rho_i \sum_{n_k, \forall k \neq i} \big[P(n_1, \dots, n_i - 1, \dots, n_m) \\ & - P(n_1, \dots, n_i, \dots, n_m)\big] \\ =& \rho_i [P(n_i - 1) - P(n_i)]\end{aligned}$$

for $j = i$. These results imply

$$\sum_{n_k, \forall k \neq i} \sum_{j=1}^{m} \rho_i (\mathbb{E}_j^{-1} - 1) P(n_1, \dots, n_m) = \rho_i [P(n_i - 1) - P(n_i)]. \tag{13}$$

Similarly, we have

$$\sum_{n_k, \forall k \neq i} \sum_{j=1}^{m} d_j (\mathbb{E}_j^{1} - 1) n_j P(n_1, \dots, n_m) = d_i [(n_i + 1) P(n_i + 1) - n_i P(n_i)]. \tag{14}$$

Subsequently, it follows that for $\ell \neq i$ and $j \in \mathcal{N}^\ell$

$$\begin{aligned}&\sum_{n_k, \forall k \neq i} (\mathbb{E}_\ell^1 \mathbb{E}_j^{-1} - 1) h(n_\ell) P(n_1, \dots, n_m) \\ &= \sum_{n_j, n_\ell} \big[h(n_\ell + 1) P(n_\ell + 1, n_i, n_j - 1) - h(n_\ell) P(n_\ell, n_i, n_j)\big] \\ &= \sum_{n_\ell} \big[h(n_\ell + 1) P(n_\ell + 1, n_i) - h(n_\ell) P(n_\ell, n_i)\big] \\ &= \sum_{n_\ell} \big[h(n_\ell + 1) P(n_\ell + 1 | n_i) \times P(n_i) - h(n_\ell) P(n_\ell | n_i) \times P(n_i)\big] \\ &= P(n_i) \big[\langle h(n_\ell) | n_i \rangle - \langle h(n_\ell) | n_i \rangle\big] = 0,\end{aligned}$$

and for $\ell = i$ and $j \in \mathcal{N}^i$,

$$\begin{aligned}&\sum_{n_k, \forall k \neq i} (\mathbb{E}_i^1 \mathbb{E}_j^{-1} - 1) h(n_i) P(n_1, \dots, n_m) \\ &= \sum_{n_j} \big[h(n_i + 1) P(n_i + 1, n_j - 1) - h(n_i) P(n_i, n_j)\big] \\ &= \sum_{n_j} \big[h(n_i + 1) P(n_i + 1, n_j) - h(n_i) P(n_i, n_j)\big] \\ &= h(n_i + 1) P(n_i + 1) - h(n_i) P(n_i),\end{aligned}$$

both of which imply that

$$\sum_{n_k,\forall k\neq i}\sum_{i=1}^{m}\sum_{j\in\mathcal{N}^i}(\mathbb{E}_i^1\mathbb{E}_j^{-1}-1)h(n_i)P(n_1,\ldots,n_m)$$
$$=\#\mathcal{N}^i\left[h(n_i+1)P(n_i+1)-h(n_i)P(n_i)\right]. \tag{15}$$

Lastly, we perform marginalisation of the last term on the right hand side of Eq. (3). It follows that for $\ell\neq i$ and $j\in\mathcal{N}^\ell$

$$\begin{aligned}&\sum_{n_k,\forall k\neq i}(\mathbb{E}_\ell^{-1}\mathbb{E}_j^1-1)h(n_j)P(n_1,\ldots,n_m)\\&=\sum_{n_j,n_\ell}\left[h(n_j+1)P(n_\ell-1,n_i,n_j+1)-h(n_j)P(n_\ell,n_i,n_j)\right]\\&=\sum_{n_j}\left[h(n_j+1)P(n_j+1,n_i)-h(n_j)P(n_j,n_i)\right]\\&=\sum_{n_j}\left[h(n_j+1)P(n_j+1|n_i)\times P(n_i)-h(n_j)P(n_j|n_i)\times P(n_i)\right]\\&=P(n_i)\left[\langle h(n_j)|n_i\rangle-\langle h(n_j)|n_i\rangle\right]=0,\end{aligned}$$

and for $\ell=i$ and $j\in\mathcal{N}^i$,

$$\begin{aligned}&\sum_{n_k,\forall k\neq i}(\mathbb{E}_i^{-1}\mathbb{E}_j^1-1)h(n_j)P(n_1,\ldots,n_m)\\&=\sum_{n_j}\left[h(n_j+1)P(n_i-1,n_j+1)-h(n_j)P(n_i,n_j)\right]\\&=\sum_{n_j}\left[h(n_j+1)P(n_j+1|n_i-1)\times P(n_i-1)-h(n_j)P(n_j|n_i)\times P(n_i)\right]\\&=\langle h(n_j)|n_i-1\rangle P(n_i-1)-\langle h(n_j)|n_i\rangle P(n_i).\end{aligned}$$

These imply that

$$\begin{aligned}&\sum_{n_k,\forall k\neq i}\sum_{i=1}^{m}\sum_{j\in\mathcal{N}^i}(\mathbb{E}_i^{-1}\mathbb{E}_j^1-1)h(n_j)P(n_1,\ldots,n_m)\\&=\sum_{j\in\mathcal{N}^i}\left[\langle h(n_j)|n_i-1\rangle P(n_i-1)-\langle h(n_j)|n_i\rangle P(n_i)\right].\end{aligned} \tag{16}$$

### 5.2. Neural network specifications

In the example of birth–death reactions in a system of connected cells, the truncation number for the protein count distribution is $N=39$. The neural network takes the marginal distributions of two adjacent cells as inputs and outputs the effective transport influx propensity into each voxel. The neural network has 3 layers with 80-20-39 neurons from input to output respectively, while the activation functions are `tanh` and `LeakyReLU` accordingly.

We opted for the three-layer architecture due to its optimal balance between simplicity and computational efficiency. Empirical evidence demonstrated that a single hidden layer effectively captures the necessary complexity for our applications. Additionally, it should be noted that methods like automated network architecture search could be employed to efficiently identify the most suitable network structure [52]. The two-cell distribution snapshots used for training are evenly distributed over time $t=20$ with time interval $t_s=0.1$. While it is possible to generate distribution snapshots from multi-cell ($>2$ cells) systems for training, we believe the incremental accuracy improvements would be marginal and thus not worth the increased computational cost. This marginal gain can be alternatively achieved by simulating additional two-cell data across a diverse range of kinetic parameter sets.

The SSA sample number is $8\times10^4$ for Figs. 2a–c. The initial condition is zero protein in all cells. The loss function is the sum of the mean squared difference between the distributions predicted by the FSP of the two-voxel GNN-MME and the SSA distribution snapshots; this sum is over all time points, kinetic parameter sets and cells. The minimisation of the loss function is performed by gradient based optimisation algorithms. Specifically, gradients are computed by automatic differentiation via packages `DiffEqSensitivity` and `Zygote`, backpropagation is used to efficiently compute the gradient of the loss function with respect to the parameters of the neural network and the `Adam` optimiser is then used to update the parameters to minimise loss. The learning rate of the optimiser is set to $\eta=0.01$ for 300 epochs.

In the example of whole cell model of a genetic negative feedback loop, the truncation number for the protein count distribution is $N=39$. For the four neural networks in Eq. (9), $f_{\theta_1,\text{intra}_M}$ and $f_{\theta_2,\text{intra}_P}$ take the marginal distributions of mRNA and protein from the same voxel as inputs and output the effective transcriptional and translational propensities respectively, while $f_{\theta_3,\text{inter}_M}$ and $f_{\theta_4,\text{inter}_P}$ take the marginal distributions of mRNA or protein of two adjacent voxels as inputs and output the effective diffusive influx propensities from the neighbouring cell. The four neural networks have 3 layers with 80-20-39 neurons from input to output respectively, while the activation functions are `tanh` and `LeakyReLU` accordingly. The two-voxel distribution snapshots are evenly distributed over time $t=60$ with time interval $t_s=0.5$, and the SSA sample number is $8\times10^4$. The initial condition is zero mRNA and protein. The loss function is similar to what was defined for the previous system. The training is performed by using the `Adam` optimiser with two stages: the first stage uses a learning rate $\eta=0.01$ for 300 epochs, while the second stage uses $\eta=0.001$ for another 300 epochs.

## Code availability

The codes, readme file and data for GNN-MME can be found at `10.5281/zenodo.7678839`. The codes are implemented by `Julia` 1.6.5 and its package `Flux` v0.12.8, `DifferentialEquations` v7.0.0, `DiffEqSensitivity` v6.66.0 and `Zygote` v0.6.33.

## CRediT authorship contribution statement

**Zhixing Cao:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Rui Chen:** Visualization, Formal analysis. **Libin Xu:** Visualization, Formal analysis. **Xinyi Zhou:** Visualization, Formal analysis. **Xiaoming Fu:** Visualization, Formal analysis. **Weimin Zhong:** Visualization, Formal analysis. **Ramon Grima:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization.

## Declaration of competing interest

The authors declare no conflict of interests.

## Data availability

https://zenodo.org/records/7678839.

## Acknowledgements

We acknowledge Rui Wang for a preliminary study exploring the viability of GNNs in solving stochastic gene expression problems. We thank Kaan Öcal and Augustinas Sukys for comments on the manuscript. W.Z. acknowledges support from the Natural Science Foundation of China (NSFC No. 61925305). Z.C. acknowledges support from NSFC No. 62073137, Shanghai Action Plan for Technological Innovation Grant (No. 22ZR1415300, 22511104000, 23S41900500), the Natural Science and Engineering Research Council of Canada's (NSERC's) Discovery Grant (RGPIN-2024-06015) and Shanghai Center of Biomedicine Development. R.G. acknowledges support from the Leverhulme Trust grant (RPG-2020-327).

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.mbs.2024.109248.

## References


[1] C.I. Maeder, et al., Spatial regulation of fus3 map kinase activity through a reaction–diffusion mechanism in yeast pheromone signalling, Nature Cell Biol. 9 (2007) 1319–1326.
[2] S. Kondo, T. Miura, Reaction–diffusion model as a framework for understanding biological pattern formation, Science 329 (2010) 1616–1620.
[3] M.B. Short, P.J. Brantingham, A.L. Bertozzi, G.E. Tita, Dissipation and displacement of hotspots in reaction–diffusion models of crime, Proc. Natl. Acad. Sci. 107 (2010) 3961–3965.
[4] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, A. Vespignani, Epidemic processes in complex networks, Rev. Modern Phys. 87 (2015) 925.
[5] G.-Q. Sun, M. Jusup, Z. Jin, Y. Wang, Z. Wang, Pattern transitions in spatial epidemics: Mechanisms and emergent properties, Phys. Life Rev. 19 (2016) 43–73.
[6] I.R. Epstein, B. Xu, Reaction–diffusion processes at the nano-and microscales, Nature Nanotechnol. 11 (2016) 312–319.
[7] V. Colizza, R. Pastor-Satorras, A. Vespignani, Reaction–diffusion processes and metapopulation models in heterogeneous networks, Nat. Phys. 3 (2007) 276–282.
[8] D. Schnoerr, R. Grima, G. Sanguinetti, Cox process representation and inference for stochastic reaction–diffusion processes, Nat. Commun. 7 (2016) 11729.
[9] S. Smith, R. Grima, Single-cell variability in multicellular organisms, Nature Commun. 9 (2018) 345.
[10] Z.R. Thornburg, et al., Fundamental behaviors emerge from simulations of a living minimal cell, Cell 185 (2022) 345–360.
[11] K. Takahashi, S. Tănase-Nicola, P.R. Ten Wolde, Spatio-temporal correlations can drastically change the response of a mapk pathway, Proc. Natl. Acad. Sci. 107 (2010) 2473–2478.
[12] R. Erban, S.J. Chapman, Stochastic Modelling of Reaction–Diffusion Processes, vol. 60, Cambridge University Press, 2020.
[13] F. Graner, J.A. Glazier, Simulation of biological cell sorting using a two-dimensional extended potts model, Phys. Rev. Lett. 69 (1992) 2013.
[14] T. Newman, R. Grima, Many-body theory of chemotactic cell–cell interactions, Phys. Rev. E 70 (2004) 051916.
[15] J.S. van Zon, P.R. Ten Wolde, Simulating biochemical networks at the particle level and in time and space: Green's function reaction dynamics, Phys. Rev. Lett. 94 (2005) 128103.
[16] S.S. Andrews, N.J. Addy, R. Brent, A.P. Arkin, Detailed simulations of cell biology with smoldyn 2.1, PLoS Comput. Biol. 6 (2010) e1000705.
[17] C. Gardiner, K. McNeil, D. Walls, I. Matheson, Correlations in stochastic theories of chemical reactions, J. Stat. Phys. 14 (1976) 307–331.
[18] D.T. Gillespie, L.R. Petzold, E. Seitaridou, Validity conditions for stochastic chemical kinetics in diffusion-limited systems, J. Chem. Phys. 140 (2014) 02B604_1.
[19] S. Smith, R. Grima, Spatial stochastic intracellular kinetics: A review of modelling approaches, Bull. Math. Biol. 81 (2019) 2960–3009.
[20] M.J. Del Razo, et al., A probabilistic framework for particle-based reaction–diffusion dynamics using classical fock space representations, Lett. Math. Phys. 112 (2022) 49.
[21] K.M. Crawford, P.C. Zambryski, Subcellular localization determines the availability of non-targeted proteins to plasmodesmatal transport, Curr. Biol. 10 (2000) 1032–1040.
[22] W.J. Lucas, et al., Selective trafficking of knotted1 homeodomain protein and its mrna through plasmodesmata, Science 270 (1995) 1980–1983.
[23] B. Munsky, M. Khammash, The finite state projection algorithm for the solution of the chemical master equation, J. Chem. Phys. 124 (2006) 044104.
[24] D. Bernstein, Simulating mesoscopic reaction–diffusion systems using the Gillespie algorithm, Phys. Rev. E 71 (2005) 041103.
[25] D.T. Gillespie, Stochastic simulation of chemical kinetics, Annu. Rev. Phys. Chem. 58 (2007) 35–55.
[26] R. Erban, S.J. Chapman, Stochastic modelling of reaction–diffusion processes: algorithms for bimolecular reactions, Phys. Biol. 6 (2009) 046001.
[27] R. Ramaswamy, I.F. Sbalzarini, Exact on-lattice stochastic reaction–diffusion simulations using partial-propensity methods, J. Chem. Phys. 135 (2011) 244103.
[28] B. Drawert, S. Engblom, A. Hellander, URDME: a modular framework for stochastic simulation of reaction-transport processes in complex geometries, BMC Syst. Biol. 6 (2012) 76.
[29] Q. Jiang, et al., Neural network aided approximation and parameter inference of non-Markovian models of gene expression, Nature Commun. 12 (2021) 2618.
[30] A. Gupta, C. Schwab, M. Khammash, DeepCME: A deep learning framework for computing solution statistics of the chemical master equation, PLoS Comput. Biol. 17 (2021) e1009623.
[31] L. Bortolussi, L. Palmieri, Deep abstractions of chemical reaction networks, in: Computational Methods in Systems Biology, 2018, pp. 21–38.
[32] A. Sukys, K. Öcal, R. Grima, Approximating solutions of the chemical master equation using neural networks, iScience 25 (2022) 105010.
[33] G. Gorin, M.T. Carilli, T. Chari, L. Pachter, Spectral neural approximations for models of transcriptional dynamics, 2022, bioRxiv 2022–06.
[34] Y. Tang, J. Weng, P. Zhang, Neural-network solutions to stochastic reaction networks, Nat. Mach. Intell. (2023) 1–10.
[35] J. Hu, et al., SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network, Nat. Methods 18 (2021) 1342–1351.
[36] J. Wang, et al., ScGNN is a novel graph neural network framework for single-cell RNA-Seq analyses, Nature Commun. 12 (2021) 1882.
[37] M.M. Li, K. Huang, M. Zitnik, Graph representation learning in biomedicine and healthcare, Nat. Biomed. Eng. (2022) 1–17.
[38] P. Reiser, et al., Graph neural networks for materials science and chemistry, Commun. Mater. 3 (2022) 93.
[39] J. Zhou, et al., Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81.
[40] Z. Wu, et al., A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (2020) 4–24.
[41] J.K. Kim, K. Josić, M.R. Bennett, The relationship between stochastic and deterministic quasi-steady state approximations, BMC Syst. Biol. 9 (2015) 1–13.
[42] Y.M. Song, H. Hong, J.K. Kim, Universally valid reduction of multiscale stochastic biochemical systems using simple non-elementary propensities, PLoS Comput. Biol. 17 (2021) e1008952.
[43] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366.
[44] C. Rackauckas, et al., Universal differential equations for scientific machine learning, 2020, arXiv preprint arXiv:2001.04385.
[45] P.W. Battaglia, et al., Relational inductive biases, deep learning, and graph networks, 2018, arXiv preprint arXiv:1806.01261.
[46] R. Grima, D.R. Schmidt, T.J. Newman, Steady-state fluctuations of a genetic feedback loop: An exact solution, J. Chem. Phys. (2012) 137.
[47] C. Jia, R. Grima, Small protein number effects in stochastic models of autoregulated bursty gene expression, J. Chem. Phys. (2020) 152.
[48] B. Alberts, et al., Molecular Biology of the Cell, seventh ed., WW Norton & Company, 2022.
[49] Z. Cao, R. Grima, Linear mapping approximation of gene regulatory networks with stochastic dynamics, Nature Commun. 9 (2018) 3305.
[50] L. Bronstein, H. Koeppl, Marginal process framework: A model reduction tool for Markov jump processes, Phys. Rev. E 97 (2018) 062147.
[51] K. Öcal, G. Sanguinetti, R. Grima, Model reduction for the chemical master equation: An information-theoretic approach, J. Chem. Phys. (2023) 158.
[52] D. Repin, T. Petrov, Automated deep abstractions for stochastic chemical reaction networks, Inform. and Comput. 281 (2021) 104788.