Learning Reaction-Diffusion Systems
with Spatial Dynamic Boltzmann Distributions

Oliver K. Ernst,1 Tom Bartol,2 Terrence Sejnowski,2,3 and Eric Mjolsness4

1Department of Physics, University of California at San Diego, La Jolla, California
2Salk Institute for Biological Studies, La Jolla, California
3Division of Biological Sciences, University of California at San Diego, La Jolla, California
4Departments of Computer Science and Mathematics, and Institute for Genomics and Bioinformatics, University of California at Irvine, Irvine, California

A common goal in machine learning is the approximation of a probability distribution by estimating interaction parameters in a graph. Distributions that evolve in both space and time appear in diverse scientific fields, but the approximation of their dynamics by a suitable conditional probability distribution is generally not straightforward. We present a shift in the learning problem for time series and spatiotemporal distributions from learning a stationary or conditional distribution to learning differential equation systems. Compared to Boltzmann machine learning, this new learning problem is derived as a variational problem for functions rather than parameters, and admits an algorithmic solution in the form of a partial differential equation constrained optimization problem rather than an unconstrained one. While this increased computational complexity prevents the method from competing with existing black box methods for learning time series, the differential equation framework instead offers a principled way to introduce problem specific physics by a suitable parameterization. One application domain for such physics-informed learning algorithms is to modeling reaction-diffusion systems. We present examples for learning chemical kinetics on a lattice, where this method offers a new take on moment closure and dimensionality reduction. We additionally highlight recent work to treat systems in continuous space, and discuss broader implications for multiscale modeling in biology.
I. INTRODUCTION

A fundamental assertion in machine learning is that data are samples of an unknown probability distribution, with the goal of estimating this distribution by a structured approach. Typically, a graphical model for the distribution is introduced and learned by determining interaction parameters between random variables, for example, by the Boltzmann machine (BM) learning algorithm [1]. In its simplest form, BM learning estimates a stationary distribution. Sequences such as time series can be learned by a number of adaptations of BM learning, such as estimating conditional distributions [25], adaptations of recurrent networks [26], or otherwise by state-space models [3] [13], switching state-space models [14], hidden Markov models [7], or a multitude of other methods [7] [22]. It is an open challenge to adapt these methods to distributions that evolve in space as well as time, i.e. spatiotemporal distributions. A further challenge in many scientific applications is that neither time nor space are discretized, i.e. the system is described by random variables representing space continuously, and varying continuously in time. Examples of such applications include: modeling spatial chemical kinetics [15], tracking neural population activity [11] [21], model reduction of fluid flow [28], traffic forecasting [17], navigation of autonomous vehicles [10], and others.

In this paper we present a recent [10] learning problem for spatiotemporal distributions. For systems described by discrete random variables, differential equation (DE) systems are estimated which dictate the time-evolution of interaction parameters in a graph. This differs fundamentally from existing methods to describe time series - in the proposed approach: (1) time is not discretized, and (2) a dynamical model is introduced and estimated. Particularly for scientific applications, prior information about the system may be introduced by parameterizing the functional forms of the DEs. This allows a general PDE (partial differential equation) model to be estimated. For systems described by continuous random variables, we generalize interaction parameters to spatial interaction functions.

The algorithmic solution to this new learning problem takes the form of a PDE-constrained optimization problem. The algorithm and its derivation are closely related to BM learning, but in this case data samples are trajectories in space and time, rather than instantaneous snapshots or slices. A related framework, graph-constrained correlation dynamics (GCCD) [15], has a similar learning goal, but uses snapshots in time, and does not consider spatial systems.

Reaction-diffusion systems are an example of where parameterizations of the DE model can be inferred from relevant physics. Field-theoretic methods [3] and the chemical master equation (CME) [12] offer analytic insight into these systems. Ernst et al. [10] showed how solutions to toy problems can be used to infer parameterizations in a black box learning approach. In this paper, we focus on lattice chemical kinetics, described by the CME. We show how this machine learning problem implements a moment closure approximation, a well-studied problem in many-body systems.

In Section II dynamic Boltzmann distributions are introduced; in Section III a learning problem is formulated for the DEs governing their time-evolution. In Section IV examples are presented for learning chemical kinetics on lattices. In Section V the generalization to spatial dynamic Boltzmann distributions is highlighted, and their relevance to model reduction of complex systems is discussed.

II. DYNAMIC BOLTZMANN DISTRIBUTIONS

A Boltzmann distribution on a state \( v = \{v_1, \ldots, v_N\} \) of \( N \) discrete random variables is of the form:

\[
\hat{p}(v) = \frac{1}{Z} \exp[-E(v)],
\]

(1)

where \( Z \) is the normalizing partition function, and \( E(v) \) is an energy function, typically defined by a chosen Markov random field (MRF). For example, a Boltzmann machine (BM) [1] is a binary MRF, where binary units update their state based on a bias and pairwise connections to other units. A MRF where all variables \( v \) are driven by data is fully visible; otherwise units \( h \) which are not driven by data are denoted as hidden. For example, a typical energy function is of the form:

\[
E(v, h, \theta) = - \sum_i b_i v_i - \sum_j b'_j h_j - \sum_{\{i_1, i_2\}} J_{i_1, i_2} v_{i_1} v_{i_2} - \sum_{\{v_j\}} W_{i,j} v_i h_j,
\]

(2)

where the sums \( \ldots \) are determined by the graph edges, and \( \theta \) denotes the complete set of \( K \) interaction parameters in the graph. This defines a joint distribution over \( v \) and \( h \):

\[
\hat{p}(v, h | \theta) = \frac{1}{Z(\theta)} \exp[-E(v, h, \theta)].
\]

(3)
Boltzmann distributions are maximum entropy (MaxEnt) distributions, where each parameter \( \theta_k \) controls a corresponding moment \( \tilde{\mu}_k \), given by

\[
\tilde{\mu}_k = \frac{\partial \ln Z(\theta)}{\partial \theta_k}.
\] (4)

To describe time-evolving distributions, define a dynamic Boltzmann distribution as one of the form

\[
\tilde{p}(v, h|\theta(t)) = \frac{1}{Z(\theta(t))} \exp[-E(v, h, \theta(t))],
\] (5)

where we have escalated to time-dependent interaction parameters. For the example (2):

\[
E(v, h, \theta(t)) = -\sum_i b_i(t)v_i - \sum_j b_j(t)h_j - \sum_{\{i,j\}} J_{i,j}(t)v_i'v_j' - \sum_{\{i,j\}} W_{i,j}(t)v_i' h_j.
\] (6)

### III. A LEARNING PROBLEM FOR DYNAMIC BOLTZMANN DISTRIBUTIONS

**Algorithm 1** Batch Gradient Descent for Learning Dynamics

1. **Initialize**
2. Grid of values \( \theta \) over which to solve for the function values \( \{F_k(\theta)\}_{k=1}^K \).
3. Time interval \([t_0, T]\), a formula for the learning rate \( \lambda \), batch size \( \eta \), no. CD steps \( \zeta \).
4. **while** not converged **do**
5. Initialize \( \Delta F_k(\theta) = 0 \) for all \( k = 1, \ldots, K \) at all grid points \( \theta \).
6. **for** sample in batch **do**
   7. **\( \triangleright \)** Generate trajectory in reduced space \( \theta \):
   8. Solve the PDE constraint (7) for a given IC over \( t_0 \leq t \leq T \) for all \( k \).
   9. Evaluate moments \( \mu_k(t) \) of the data for all \( k, t \).
   10. **\( \triangleright \)** Awake phase:
   11. Evaluate moments \( \tilde{\mu}_k(t) \) of the Boltzmann distribution by CD sampling for all \( k, t \).
   12. **\( \triangleright \)** Asleep phase:
   13. Evaluate \( \tilde{\mu}_k(t) \) of the Boltzmann distribution by CD sampling for all \( k, t \).
   14. **\( \triangleright \)** Evaluate the objective function:
   15. Update \( \Delta F_k(\theta) \) as the cumulative moving average of (6) over the batch.
   16. **\( \triangleright \)** Update to decrease objective function:
   17. \( F_k(\theta) \rightarrow F_k(\theta) - \lambda \Delta F_k(\theta) \) for all \( k, \theta \).

Given a training dataset, the BM learning algorithm determines the interaction parameters such that the stationary distribution (3) is the MaxEnt dist. consistent with the moments in the dataset. To learn a time-evolving dataset, we seek the distribution that is at all times the MaxEnt solution. Introduce for each interaction parameter \( \theta_k \) a time-evolution function \( F_k \) forming an autonomous DE system:

\[
\frac{d}{dt} \theta_k(t) = F_k(\theta(t)) \quad \text{with I.C.:} \quad \theta_k(t_0) = \eta_k.
\] (7)

In section [V] this model is further extended to a functional form for PDEs. In practice, the functions \( F \) on the right may also be restricted in their arguments to reduce computational complexity.

To formulate a learning problem for \( F \), let the unknown distribution from which training samples are assumed to be drawn from be denoted by \( \tilde{\rho} \). Define as the action the KL-divergence between the true and reduced models, \( p \) and \( \tilde{\rho} \), over all times:

\[
S = \int_{t_0}^{\infty} dt \mathcal{D}_{KL}(p||\tilde{\rho}) \quad \text{where} \quad \mathcal{D}_{KL}(p||\tilde{\rho}) = \sum_z p(z) \ln \frac{p(z)}{\tilde{\rho}(z)}.
\] (8)

The variational problem for the functions can be derived [10] as

\[
\frac{\delta S}{\delta F_k(\theta)} = \sum_{k'=1}^K \int_{t_0}^{\infty} dt' \left( \mu_{k'}(t') - \tilde{\mu}_{k'}(t') \right) \frac{\delta \theta_{k'}(t')}{\delta F_k(\theta)} = 0.
\] (9)
where $\mu_{k}(t')$ and $\tilde{\mu}_{k}(t')$ are averages taken over to $p$ and $\tilde{p}$ at time $t'$. The variational term on the right weights the effect that a perturbation in the DE system [19] at a point has on the solution trajectory. At one instant in time, the unweighted term in the integral is the learning rule of BM learning. One method to evaluate such weighting terms is by the PDE system [10]

$$
\frac{d}{dt} \left( \frac{\delta \theta_{k}(t')}{\delta F_{k}(\theta)} \right) = \sum_{i=1}^{K} \frac{\partial F_{k}(\theta(t'))}{\partial \theta_{i}(t')} \frac{\delta \theta_{i}(t')}{\delta F_{k}(\theta)} + \delta_{k,k'} \delta(\theta - \theta(t')) \quad \text{with:} \quad \frac{\delta \theta_{k}(t' = t_{0})}{\delta F_{k}(\theta)} = 0, \tag{10}
$$

where the multivariate Dirac delta is $\delta(\theta - \theta(t')) = \prod_{k=1}^{K} \delta(\theta_{k} - \theta_{k}(t'))$.

The variational problem can be solved algorithmically by a PDE-constrained optimization problem: solve [9, 10] subject to the PDE-constraint [7]. Algorithm [1] is such an algorithm using stochastic gradient descent. Standard algorithmic improvements are possible, such as accelerated gradient descent methods. More fundamental improvements may come from algorithms specific to PDE-constrained optimization problems, such as sequential quadratic programming or adjoint methods.

### IV. LEARNING REACTION-DIFFUSION SYSTEMS ON LATTICES

The state of a reaction-diffusion system at some time is described by $n$ particles of species $\alpha$ located at positions $x$ in generally continuous 3D space. To make an explicit connection to binary random variables, we consider a simpler model of particles hopping on a lattice in the single-occupancy limit. To simulate such a system, we follow Takayasu and Tretyakov [27] for a lattice-based variant of the popular Gillespie stochastic simulation algorithm (SSA) [14]: at each timestep, perform unimolecular reactions following the Gillespie SSA, then iterate over all particles in random order; for each:

1. Hop to a neighboring site, chosen to random site with equal probability.
2. If the site is occupied, a reaction occurs with some probability, or the move is rejected.

The lattice on which particles hop is the visible part of the MRF. Let the vector of possible species be $s$ of size $M$ in some arbitrary ordering (excluding $\emptyset$ to denote an empty site). Assign a unique index $i$ to each of the $N$ sites in the lattice. Spins at a site $i$ are now multinomial units, represented as a vector $v_{i}$ of length $M$ where entries $v_{i,\alpha} \in \{0,1\}$ for $\alpha = 1, \ldots, M$ denote the absence or presence of a particle of species $s_{\alpha}$. The single-occupancy limit corresponds to the implicit constraint that the vectors are of length 1, i.e. $\sum_{\alpha=0}^{M} v_{i,\alpha} = 1$, where $\alpha = 0$ denotes an empty site. The matrix $V$ of size $N \times M$ describes the state of the visible part of the MRF, where rows denote lattice sites.

Likewise introduce hidden layer species $s'$ of size $M'$, which may be different from $s$. Indexing all hidden sites as $j = 1, \ldots, N'$, hidden unit vectors are $h_{j}$ of length $M'$. The state of the hidden units is $H$ of size $N' \times M'$, with the single occupancy constraint as before. Interaction parameters $\theta(t)$ may also be species-dependent (excluding $\emptyset$). The dynamic Boltzmann dist. becomes: $\hat{p}(V, H, \theta(t)) = \exp[-E(V, H, \theta(t))]/Z(\theta(t))$. For example, the energy function [5] above becomes:

$$
E(V, H, \theta(t)) = -\sum_{i=1}^{N} \sum_{\alpha=1}^{M} b_{i,\alpha}(t) v_{i,\alpha} - \sum_{j=1}^{M'} \sum_{\beta=1}^{M'} b'_{j,\beta}(t) h_{j,\beta}
- \sum_{\{i,j\}} \sum_{\alpha,\beta} W_{i,j,\alpha,\beta}(t) v_{i,\alpha} h_{j,\beta} - \sum_{\{i_{1},i_{2}\}} \sum_{\alpha_{1},\alpha_{2}} J_{i_{1},i_{2},\alpha_{1},\alpha_{2}}(t) v_{i_{1},\alpha_{1}} v_{i_{2},\alpha_{2}}. \tag{11}
$$

#### A. Learning hidden layers for moment closure

A typical problem in many-body systems is the appearance of a hierarchy of moments, where the time-evolution of a given moment depends on higher order moments. Moment closure approximations terminate this infinite hierarchy at some finite order. In this section, we develop the perspective of the learning problem [9] as a closure approximation using a simple pedagogical example. We note some similarity to a previously proposed closure scheme [24].

Consider a bimolecular-annihilation process on a 1D lattice of length $N$, where particles of a single species $A$ hop and react according to $A + A \rightarrow \emptyset$ with probability $p_{r}$. The time-evolution of the moments may be derived from the
FIG. 1. Left: 1D lattice with one hidden layer. In this pedagogical example, $W$ is not matrix but a single parameter. Right: Time-evolution function diagram representation of (14).

FIG. 2. Learned time-evolution functions for (13, 14). Top row: Fully visible model. Left: Training set of initial points $(b, J, K)$ (cyan) sampled evenly in $[-1, 1]$. Stochastic simulations for each initial point are used as training data (learned trajectories shown in black, endpoints in magenta). Middle: Functions learned on grids of $20 \times 20 \times 20$ points, evenly spaced from $[-1.5, -1.5]$ in each dimension. Right: Percent error in $\langle \sum_i v_i v_{i+1} v_{i+2} v_{i+3} \rangle(t)$ for a set of test trajectories. Bottom row: Hidden layer model. Initial points are generated by BM learning the points of (a). The functions are learned on grids of $20 \times 20 \times 20$ points, evenly spaced over $b \in [-3, 2], b' \in [-3, 3.5], W \in [-4, 3]$.

The CME - for example, the first two are:

$$
\frac{d}{dt} \langle \sum_i v_i \rangle = -2k_r \langle \sum_i v_i v_{i+1} \rangle, \\
\frac{d}{dt} \langle \sum_i v_i v_{i+1} \rangle = 2D \langle \sum_i v_i v_{i+2} \rangle - 2k_r \langle \sum_i v_i v_{i+1} v_{i+2} \rangle + (k_r - 2D) \langle \sum_i v_i v_{i+1} \rangle,
$$

(12)

where $k_r$ is the reaction rate and $D$ the diffusion constant. The simplest graph to capture such observables is a fully visible MRF, i.e. a 1D Ising model including interactions up to some order. For example, including third order interactions, let:

$$
E(v, b(t), J(t), K(t)) = -b(t) \sum_{i=1}^N v_i - J(t) \sum_{i=1}^{N-1} v_i v_{i+1} - K(t) \sum_{i=1}^{N-2} v_i v_{i+1} v_{i+2},
$$

(13)

Hidden layers can potentially improve upon a fully visible closure model:

1. In any closure scheme, moments beyond a certain order are not captured explicitly by the model, such that their approximation may be poor. Hidden layers can be used to incorporate information about which higher order moments are relevant to the dataset.

2. Two distinct states having the same lower order moments are indistinguishable in the reduced model. The connectivity of the hidden layers in the graph may be chosen based on the DEs derived from the CME to separate such states, even in a low dimensional model.

3. The number of higher-order terms appearing on the right of (12) grows with the order on the left. This problem
FIG. 3. Rössler oscillator on a 3D lattice. (a) Snapshots of a stochastic simulation on a $10 	imes 10 	imes 10$ lattice ($A, B, C$ in pink, orange, cyan). (b) Section of graph to learn and time-evolution function diagram of (16). Circles denote visible sites as in (a); squares denote hidden units with NN connectivity (biases not shown). (c)-(e) Moments from a single simulation over 500 timesteps. The characteristic attractor (c) is a stochastic version of the well-known deterministic model.

is compounded if species labels are included. Hidden layers may be used to approximate such higher order interactions with fewer parameters.

A simple graph to learn using hidden layers is shown in Figure 1, corresponding to:

$$E(v, h, b(t), W(t), b'(t)) = -b(t) \sum_{i=1}^{N} v_i - b'(t) \sum_{j=1}^{N-1} h_j - W(t) \sum_{i=1}^{N} \sum_{j=i+1}^{N} v_i h_j,$$

$$\frac{d}{dt} \gamma = F_{\gamma}(b, b', W) \quad \text{for} \quad \gamma = b, b', W. \quad (14)$$

The time-evolution functions for (13) and (14) are learned using Algorithm 1 and compared in Figure 2. As training data, 50 points $(b, J, K)$ are sampled evenly over $(b, J, K) \in [-1, 1]^3$. Each point corresponds to an initial distribution (13), from each of which 100 lattices of length $N = 1000$ are sampled. Each lattice is simulated for 100 timesteps of size $\Delta t = 0.01$ with $p_r = 0.01$. These trajectories are pooled for Algorithm 1, with parameters: for the visible model: batch size $\eta = 10$, no. CD (contrastive divergence) steps $\zeta = 1$, learning rate $\lambda = 0.001$, for 1000 optimization steps, and the same for the hidden model with $\lambda = 0.01$ for 500 steps. The source terms in (10) are approximated by normalized Gaussians with variance equal to the grid spacing in each dimension. To improve convergence, the Nesterov method was used. Figure 2 shows the learned time-evolution functions.

After sampling the learned models, both show close agreement to the moments from stochastic simulation. As a test set, we sample points $(b, J, K)$ as before and evolve these using the learned DE system. Figure 3 shows the percent error for $(\sum v_i v_{i+1} v_{i+2} v_{i+3}) (t)$ compared against stochastic simulation, showing to what accuracy different regions of parameter space have been learned. Note that in general, the mapping (in a MaxEnt sense) of states described by observables to states in interaction parameter space needs not be injective. Here for example, states with $N/2$ particles can map to $(b, b', W) = (0, -\infty, -\infty)$, regardless of higher order moments. The choice of graph must be uniquely tailored to the dataset.

B. Rössler oscillator

The Williamowski-Rössler oscillator system [6] is a chemical version of a spiral oscillator in three species. The original formulation requires additional species that are fixed at constant concentration. Recent work [4] has shown that these can be incorporated into pseudo-first order reaction rates. The oscillator for the species $A, B, C$ is dictated by the reaction system:

$$A \xrightleftharpoons{p_1} 2A \quad A + B \xrightarrow{p_2} 2B \quad A + C \xrightarrow{p_3} \emptyset \quad B \xrightarrow{k_2} \emptyset \quad C \xrightarrow{k_3} 2C \quad (15)$$

where the unimolecular reaction rates used are $k_1 = 30$, $k_2 = 10$, $k_3 = 16.5$ (arbitrary units), and the probabilities for bimolecular reactions are $p_1 = 0.1, p_2 = 0.4, p_3 = 0.24, p_4 = 0.36$. We simulate this system on a 3D lattice of size
10 × 10 × 10 sites in the single occupancy limit as before. Figure 3 shows snapshots of such a stochastic simulation. Panel (c) in particular shows the characteristic shape of the Rössler oscillator, with further structures evident in the NNs shown in (d),(e). In chaotic systems, two close initial states will diverge over their long term time-evolution. The challenge for the learning problem is to separate the DEs for these two initial conditions, despite the stochasticity in the dataset, and the dependence on higher order correlations, i.e. moment closure.

Let the visible part of the graph be the lattice of Figure 3(a). The hidden part of the graph implements NN connections between sites as shown in Figure 3(b). Introduce a single hidden species $X$, then let:

\[
E(V, H, \theta(t)) = - \sum_{i \in \text{visible}} \sum_{\alpha \in \{A, B, C\}} b_\alpha v_{i,\alpha} - \sum_{j \in \text{hidden}} b_X h_{j,X} - \sum_{\{i,j\}} \sum_{\alpha \in \{A, B, C\}} W_{\alpha,X} v_{i,\alpha} h_{j,X},
\]

\[
\frac{d}{dt} \gamma = F_\gamma(W_{AX}, W_{BX}, W_{CX}) \quad \text{for} \quad \omega = b_A, b_B, b_C, b'_X, W_{AX}, W_{BX}, W_{CX}.
\]

The algorithm must therefore learn non-intersecting trajectories in $(W_{AX}, W_{BX}, W_{CX})$-space. For training, stochastic simulations are generated from an initial state defined by (16), with parameters $b_A, b_B, b_C = -1$, otherwise zero. Simulations are run for a single cycle of 60 timesteps of size $\Delta t = 0.01$. Figure 5(a) shows the relaxation of the distribution to equilibrium [2].

The time integral in (9) poses a challenge for learning periodic trajectories. One approach is to split the trajectories into aperiodic segments, which are combined as a single training dataset. An alternative on-line approach is to incorporate an exponential time decay factor into the update rule:

\[
\sum_{k'=1}^{K} \int_{t_0}^{\infty} dt' \left( \mu_{k'}(t') - \tilde{\mu}_{k'}(t') \right) \frac{\delta \theta_{k'}(t')}{\delta F_k(\theta)} e^{-\xi(t'-t_0)} = 0,
\]

where $t_0, \xi$ may be varied. We found the fastest and most consistent convergence by combining both strategies, first dividing the simulated trajectories into segments of length 10 timesteps. After using BM learning to determine initial conditions, we use Algorithm 1 on this dataset with parameters: $\zeta = 1, \lambda = 0.001$, and large batch size $\eta = 300$ but for few optimization steps 20. We used the learned functions as initialization for the second method, where $\xi = 1$ and $t_0$ is slowly varied as $t_0/\Delta t \to t_0/\Delta t + 10$ every 20 optimization steps. The parameters used were: $\zeta = 1, \lambda = 0.001, \eta = 50$, for 100 optimization steps.

Figure 4 shows the learned functions. The trajectory resembles one loop of a helix, demonstrating that the hidden layers have separated the overlapping trajectories of Figure 3. Figure 5(b) compares the observables over this first cycle. Higher order correlations such as nearest neighbors are reasonably approximated, even though they are not explicitly captured by the model. A deeper network will likely further improve this convergence.

**FIG. 4.** Learned time-evolution functions [16]. 20 evenly spaced grid points are used in each dimension $W_{AX} \in [-0.6, 0.4], W_{BX} \in [-0.1, 0.7], W_{CX} \in [-0.1, 0.4]$. Functions are learned by first learning piecewise linear segments, and then using these as the initial condition for (17).
V. SPATIAL DYNAMIC BOLTZMANN DISTRIBUTIONS

The goal of this section is to highlight recent work [10] on a learning framework for spatially continuous systems. Again, we focus on reaction-diffusion systems as an illustrative example. Return to the state vector described by \((n, \alpha, x)\). Introduce \(k\)-particle interaction functions \(\nu_k(\alpha_{(i)}^n, x_{(i)}^n, t)\), where \((i)^n_k\) denotes any ordered subset of \(k\) indexes with each index in \(\{1, \ldots, n\}\). Given a set of such interaction functions \(\nu_k^{K}\) up to cutoff order \(K\), the spatial dynamic Boltzmann distribution is:

\[
\tilde{p}(n, \alpha, x|\nu, t) = \frac{1}{Z[\nu]} \exp \left[ -\sum_{k=1}^{K} \sum_{(i)^n_k} \nu_k(\alpha_{(i)}^n, x_{(i)}^n, t) \right],
\]

where the sum over \((i)^n_k\) iterates over unique \(k\)-th order interactions between \(n\) particles. This is a MaxEnt distribution as before, where interaction functions \(\nu_k(\alpha_{(i)}^n, x_{(i)}^n, t)\) control moments

\[
\mu_k(\alpha_{(i)}^n, x_{(i)}^n, t) = \sum_{\nu=0}^{\infty} \sum \int dx' p(n', \alpha', x', t) \sum \delta(x_{(i)}^n - x'_{(j)}^n') \delta(\alpha_{(i)}^n - \alpha'_{(j)}^n'),
\]

that is, the average number of \(k\)-sized tuplets of particles of species \(\alpha_{(i)}^n\) at locations \(x_{(i)}^n\). A more general functional model may be proposed for the interaction functions:

\[
\frac{d}{dt} \nu_k(\alpha_{(i)}^n, x_{(i)}^n, t) = \mathcal{F}_k[\nu(\alpha, x, t)].
\]

where \(\nu(\alpha, x, t)\) denotes all possible functions \(\nu\) up to order \(K\) of all possible arguments. Here, \(\mathcal{F}\) is a global functional, i.e. its arguments are not restricted to those appearing on the left.

By a suitable parameterization of \(\mathcal{F}\), a similar learning to [9] can be obtained. Ernst et al. [10] explores such parameterizations for chemical kinetics, and we refer to this work for numerical experiments. If \(\mathcal{F}\) is parameterized by ordinary functions indexed by \(\kappa\), then a general learning rule is of the form:

\[
\sum_{k'=1}^{K} \int \text{d}x' \int_0^\infty \text{d}t' \left( \mu_{k'}(\alpha', x', t') - \tilde{\mu}_{k'}(\alpha', x', t') \right) R_{k', \kappa}(\alpha', x', t', \{\nu(\alpha, x, t)\}) = 0,
\]

where \(R\) denotes the variational term of \(\nu_{k'}(\alpha', x', t')\) with respect to the \(\kappa\)-th ordinary function. To determine \(R\), further PDE systems may be derived, similar to [10].

VI. DISCUSSION

We have presented a learning problem for spatiotemporal distributions that estimates a differential equation system, rather than a stationary or conditional distribution as done in existing BM-related algorithms. The choice for the

![Fig. 5](image)

(a) Relaxation to a stationary distribution, indicated by the convergence of the means from averaging over 100 stochastic simulations. (b) Learned mean number of A, B, C and neighboring pairs of A, B and B, B for the first oscillation, obtained by sampling [16], and compared to (a).
dynamic model to be estimated can be tuned for different modeling applications, such as moment closure approximations for chemical systems as illustrated.

We emphasize the limitations of Algorithm 1, in particular the complexity of solving a PDE-constrained optimization problem. Here the number of grid points scales as $d^n$, where $d$ is the dimension of $F$ and $n$ is the number of points in each dimension. It is therefore infeasible to learn functions consisting of a large number of arguments in this black-box approach.

As a result, Algorithm 1 in its current form is not a replacement for existing generic methods for modeling time series. Rather, we view this as a formalism for incorporating relevant physics into machine learning by introducing further parameterizations into the differential equations. Parameterizations have also been used in regular BM learning methods, e.g. factoring weights \[23\]. For deep architectures, recycling the same time-evolution functions across multiple layers may be effective, similar to convolution layers in convolutional neural networks.

We view the present work as progress toward linking models across scales in biology \[18\]. Reaction-diffusion systems illustrate many of the common problems in this field. While much machinery (CME or field-theoretic methods) exists to formulate problems for observables, their solution is non-trivial in most applications. Even without analytic challenges such as moment closure, the numerical solution of PDE systems is difficult for systems with high spatial organization, or where interactions with other scales (e.g. molecular dynamics) or physics (e.g. electrodiffusion) become relevant. Spatial dynamic Boltzmann distributions and their learning rule \[21\] may abstract much of these non-trivial interactions. Similar approaches in other spatial modeling applications may prove equally promising.

[1] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science, 9*(1):147–169, 1985.
[2] VS Anishchenko, TE Vadivasova, GI Strelkova, and GA Okrokvertskhov. Statistical properties of dynamical chaos. *Mathematical Biosciences and Engineering: MBE, 1*(1):161, 2004.
[3] Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367, 2015*.
[4] Giovanni Bellesia and Benjamin B. Bales. Population dynamics, information transfer, and spatial organization in a chemical reaction network under spatial confinement and crowding conditions. *Phys. Rev. E, 94*:042306, Oct 2016.
[5] Christopher Bishop. *Pattern recognition and machine learning*. Springer, New York, 2006.
[6] Willamowski K. D. and Rössler O. E. Irregular oscillations in a realistic abstract quadratic mass action system. *Zeitschrift für Naturforschung A, 35*:317, November 1980.
[7] Sakyasingha Dasgupta and Takayuki Osogami. Nonlinear dynamic Boltzmann machines for time-series prediction. *AAAI Conference on Artificial Intelligence, 2017*.
[8] M Doi. Second quantization representation for classical many-particle system. *Journal of Physics A: Mathematical and General, 9*(9):1465, 1976.
[9] Stuart Dreyfus. The numerical solution of variational problems. *Journal of Mathematical Analysis and Applications, 5*(1):30–45, 1962.
[10] Oliver K. Ernst, Thomas Bartol, Terrence Sejnowski, and Eric Mjolsness. Learning dynamic Boltzmann distributions as reduced models of spatial chemical kinetics. *arXiv:1803.01063, 2018*.
[11] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neuronal population models through nonlinear embeddings. *In Advances in Neural Information Processing Systems, pages 163–171, 2016*.
[12] C. W. Gardiner. *Stochastic methods: A handbook for the natural and social sciences*. Springer, Berlin, 2009.
[13] Zoubin Ghahramani and Geoffrey E. Hinton. Variational learning for switching state-space models. *Neural Computation, 12*(4):831–864, 2000.
[14] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry, 81*(25):2340–2361, 1977.
[15] Todd Johnson, Tom Bartol, Terrence Sejnowski, and Eric Mjolsness. Model reduction for stochastic CaMKII reaction kinetics in synapses by graph-constrained correlation dynamics. *Physical Biology, 12*(4):045005–045005, 07 2015.
[16] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal, 1*(1):1, July 2014.
[17] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *In International Conference on Learning Representations, 2018*.
[18] Eric Mjolsness. Prospects for declarative mathematical modeling of complex biological systems. *arXiv:1804.11044, 2018*.
[19] Note1. This resembles variational problems found in early work on optimal control theory, see e.g. Dreyfus \[9\].
[20] Note2. An n-vector model in statistical mechanics.
[21] Cian O’Donnell, J. Tiago Gonçalves, Nick Whiteley, Carlos Portera-Cailliau, and Terrence J. Sejnowski. The population tracking model: A simple, scalable statistical model for neural population data. *Neural Computation, 29*(1):50–93, March 2016.
[22] Takayuki Osogami and Makoto Otsuka. Seven neurons memorizing sequences of alphabetical images via spike-timing dependent plasticity. *Scientific Reports, 5*:14149, 09 2015.
[23] Marc'Aurelio Ranzato, Alex Krizhevsky, and Geoffrey Hinton. Factored 3-way restricted Boltzmann machines for modeling natural images. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 621–628, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[24] Patrick Smadbeck and Yiannis N. Kaznessis. A closure scheme for chemical master equations. *Proceedings of the National Academy of Sciences*, 110(35):14261–14265, 2013.

[25] Ilya Sutskever and Geoffrey Hinton. Learning multilevel distributed representations for high-dimensional sequences. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 548–555, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

[26] Ilya Sutskever, Geoffrey E Hinton, and Graham W. Taylor. The recurrent temporal restricted Boltzmann machine. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1601–1608. Curran Associates, Inc., 2009.

[27] H. Takayasu and A. Yu. Tretyakov. Extinction, survival, and dynamical phase transition of branching annihilating random walk. *Phys. Rev. Lett.*, 68:3060–3063, May 1992.

[28] Kun Zhao, Takayuki Osogami, and Rudy Raymond. Fluid simulation with dynamic Boltzmann machine in batch manner. In *NIPS Time Series Workshop*, Long Beach, CA, USA, December 2017.