FITTING STOCHASTIC LATTICE MODELS USING APPROXIMATE GRADIENTS

Jan Schering[1], Sander Keemink[2,*], and Johannes Textor[1,*]

[1]Institute for Computing and Information Sciences, Radboud University
[2]Donders Institute for Brain, Cognition and Behavior, Department of AI, Radboud University
[*]These authors contributed equally.

KEYWORDS

Simulation, Machine Learning, Lattice Models, Optimization, Approximate gradient methods.

ABSTRACT

Stochastic lattice models (sLMs) are computational tools for simulating spatiotemporal dynamics in physics, computational biology, chemistry, ecology, and other fields. Despite their widespread use, it is challenging to fit sLMs to data, as their likelihood function is commonly intractable and the models non-differentiable. The adjacent field of agent-based modelling (ABM), faced with similar challenges, has recently introduced an approach to approximate gradients in network-controlled ABMs via reparameterization tricks. This approach enables efficient gradient-based optimization with automatic differentiation (AD), which allows for a directed local search of suitable parameters rather than estimation via black-box sampling. In this study, we investigate the feasibility of using similar reparameterization tricks to fit sLMs through backpropagation of approximate gradients. We consider three common scenarios: fitting to single-state transitions, fitting to trajectories, and identification of stable lattice configurations. We demonstrate that all tasks can be solved by AD using three example sLMs from sociology, biophysics, and physical chemistry. Our results show that AD via approximate gradients is a promising method to fit sLMs to data for a wide variety of models and tasks.

INTRODUCTION

Stochastic lattice models (sLMs) are computational tools for simulating spatiotemporal dynamics in both out-of-equilibrium and stable systems (Haselwandter and Vvedensky, 2008). They are applied in a wide range of scientific fields including hydrodynamics (Chopard and Droz, 2005), molecular morphogenesis (Dab et al., 1991; Malevanets and Kapral, 1997), and cell biology (Szabó and Merks, 2013; Wortel et al., 2021b). Replicating natural phenomena in silico via sLMs requires calibrating their parameters to data. Three problems make this task challenging. First, sLMs are typically not analytically tractable, precluding the computation of likelihood functions. Second, black-box optimization methods developed for deterministic functions (e.g., covariance matrix adaptation (Hansen et al., 1995)) cannot be easily applied (Beyer et al., 2002). Finally, sLMs are discrete and hence non-differentiable, preventing gradient computation.

Motivated by similar challenges, the adjacent field of agent-based modelling has investigated methods of constructing differentiable agent-based models (ABMs) that are amenable to efficient gradient-based optimization via automatic differentiation (AD) (Andelfinger, 2023; Chopra et al., 2022). An especially promising method is the combination of reparameterization tricks (Kingma and Welling, 2013) with straight-through estimators (STEs) to attain approximate gradient estimates from non-differentiable, stochastic models (Chopra et al., 2022). Reparameterization tricks enable differentiating through stochastic models with respect to some parameters of interest. STEs approximate the derivatives of discrete operations during backpropagation while upholding discreteness during simulation (Bengio et al., 2013). Combining the methods provides a way to estimate the gradient of otherwise non-differentiable functions.

A potential benefit over the state-of-the-art gradient-free Bayesian methods used for parameter inference (Alamoudi et al., 2023; Wang et al., 2022) is that it allows for a local directed search for suitable parameters, rather than estimating entire parameter distributions. The gradient information guides this local search, whereas gradient-free methods such as approximate Bayesian computation (ABC) (Sisson et al., 2018; Mengersen et al., 2013) rely on black-box sampling. Thus, gradient-based optimization could converge faster and result in better estimates, especially for higher-dimensional models (Andelfinger, 2023).

Current research on gradient-based optimization of ABMs has largely focused on network-controlled models (Chopra et al., 2022; Quera-Bofarull et al., 2023b; Andelfinger, 2023). This class of ABMs is closely related to sLMs, though some key differences make the transfer of the method challenging. In ABMs, differentiable neural networks are used to learn the behavior

Communications of the ECMS, Volume 38, Issue 1,
Proceedings, ©ECMS Daniel Grzonka, Natalia Rylko,
Grazyna Suchacka, Vladimir Mityushev (Editors) 2024
ISBN: 978-3-937436-84-5/978-3-937436-83-8(CD) ISSN 2522-2414

of an agent within an environment. In sLMs, however, we calibrate the parameters of both the agent and the environment itself, requiring approximate end-to-end differentiability of the model. If successful, transferring the approach to the domain of sLMs can similarly improve the efficiency of model calibration to data.

In this preliminary work, we provide a clear first proof-of-concept of calibrating sLMs to data through AD. We consider a range of model types and typical calibration tasks and solve them using AD via approximate gradients. **Our contributions** in this work are the following, We:

1. identify and specify three common types of sLM calibration tasks.
2. extend the use of reparameterization tricks from network-based ABMs to the sLM framework.
3. implement three different sLMs from different domains in a differentiable fashion using PyTorch.
4. show the feasibility of AD via approximate gradients for the identified tasks by performing AD on models taken from sociology, biophysics, and physical chemistry.

RELATED WORKS

Historically, derivative-free gradient estimators have been used to circumvent non-differentiability. The simplest derivative-free method to estimate gradients is finite-differencing (Shi et al., 2021). Sumata et al. (2013) use finite-differencing to calibrate an sLM of sea ice. However, finite-difference schemes can suffer from an unboundedly high variance when applied to stochastic models (Arya et al., 2022).

The currently most prominent class of methods for parameter estimation of sLMs is likelihood-free Bayesian inference. At the forefront is the ABC framework (Alamoudi et al., 2023; Wang et al., 2022), which constructs a distance measure between the summary statistics of the data and then performs Bayesian inference via Monte Carlo sampling (Sisson et al., 2018; Mengersen et al., 2013). ABC is a powerful tool with several successful applications (Durso-Cain et al., 2021; Carr et al., 2021; Beaumont et al., 2002).

More recently, Dyer et al. (2022) discusses alternative neural simulation-based inference methods for the adjacent field of ABMs. These methods enjoy the benefit of being generally more sample-efficient (Lueckmann et al., 2021) but are less amenable to theoretical study due to their black-box nature. Specifically, the impact of model misspecification is known and studied for ABC methods Frazier et al. (2020) but less clear for neural simulation-based inference. Early studies suggest a significant deterioration in the performance of the method under model misspecification (Cannon et al., 2022). To combat this, Ward et al. (2022); Kelly et al. (2023) suggest misspecification-robust extensions to neural simulation-based inference.

Enabling the use of AD by approximating the gradient has emerged from the hypothesis that directed point estimates of the mode may be beneficial compared to estimating full distributions. To this end, Andelfinger (2023) have investigated the smoothing of discrete transformations to approximate the gradient of ABMs. While delivering promising results, the method scales exponentially due to branching effects. Further, the efficient handling of stochasticity has not yet been explored in-depth.

As an alternative, reparameterization of discrete distributions via Gumbel-softmax approximation (Jang et al., 2016) in combination with STEs has been proposed and investigated for network-controlled ABMs (Quera-Bofarull et al., 2023a; Chopra et al., 2022). Early studies on the robustness of this method indicate that despite the lack of guaranteed unbiased low-variance gradients (Huijben et al., 2022), it is practically robust enough to enable accurate inference (Quera-Bofarull et al., 2023b; Chopra et al., 2022).

In summary, a large body of recent research has been dedicated to efficient calibration methods for non-differentiable and analytically intractable models. Of growing interest are methods that are amenable to gradient-based optimization via AD. To this end, the combination of reparameterization tricks and STEs to attain approximate gradients have shown promising results for network-controlled ABMs. The application of this approach to sLMs has not previously been investigated but promises similar benefits.

APPROXIMATING THE GRADIENT OF SLMS

AD is a method to find the gradient of a model with respect to some parameter by exploiting the chain rule of derivatives. Representing the model as a computational graph, gradient information is propagated back through the graph in the form of partial derivatives. Non-differentiability arises from certain types of nodes not having a well-defined partial derivative and thus blocking the backward flow of the gradient. sLMs generally consist of four types of nodes: *continuous/discrete deterministic*, *continuous/discrete stochastic* (Fig. 1).

Continuous deterministic nodes are differentiable, while the other three types block the backward flow of the gradient as shown on the left side in Fig. 1. We can assume sLMs to mostly consist of differentiable nodes with some key non-differentiable blocking nodes. The challenge is to enable the backward flow of gradient information through the blocking nodes.

Dealing with discreteness

sLMs commonly perform some form of discretization to determine the new state of a given lattice site. The derivative of every discrete transformation is 0 everywhere except for at the thresholds between categories, thus no information passes through the derivative.

STEs (Bengio et al., 2013) bypass the ill-defined partial derivative of discrete transformations by replacing it during the backward pass with an approximate derivative function (the STE). The simplest STE is the

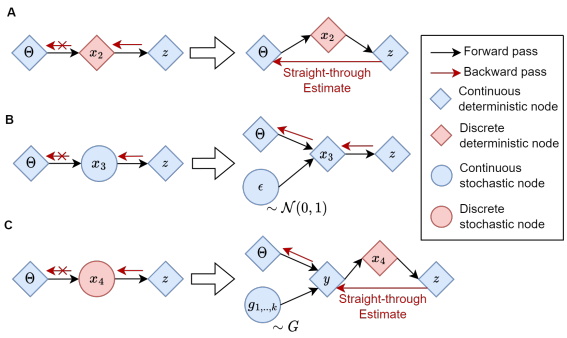

Figure 1: Gradient flow can be re-enabled with a combination of reparameterization and STEs.
A We use an STE to estimate the gradient for discrete deterministic nodes. **B** We apply the reparameterization trick to differentiate through continuous stochastic nodes. **C** Gumbel-softmax reparameterization combined with STEs estimates gradients for stochastic discrete nodes.

identity function. The viability of identity STEs has been shown in the context of spiking neuron models (Tang et al., 2022). Thus, we choose to employ identity STEs to approximate the gradient for discrete nodes. The identity STE approach is illustrated in Fig. 1**A**.

Dealing with stochasticity

Stochastic nodes represent the sampling of variables from a distribution. In sLMs, the shape of the distributions usually depends on some subset of the model parameters Θ. However, sampling operations do not have a well-defined derivative. Hence, we cannot determine the gradient with respect to the parameters that determine the shape of the distribution.

The reparameterization trick (Kingma and Welling, 2013) removes the dependency of the sampling operation on the model parameters Θ. To achieve this, an auxiliary random variable ϵ distributed by some distribution $p(\epsilon)$ that does not depend on the model parameters Θ is introduced. Then, we rewrite $z = g(\Theta, X_t, \epsilon)$ into a deterministic transformation g of the parameters, the lattice state, and the auxiliary variable. The result is equivalent to drawing a sample from the original distribution, without the sampling depending on Θ. An illustration of this is shown in Fig. 1**B**.

The Gumbel-softmax trick

The Gumbel-softmax (GS) trick (Jang et al., 2016) combines reparameterization with STEs to estimate the gradient of discrete stochastic nodes. For reparameterization, auxiliary noise variables $g_{1,2,\ldots,n}$ are sampled from the Gumbel distribution G and added to the log-probability of each class $\alpha_{1,2,\ldots,n}$. Then, softmaxing is applied. The result is a continuous approximation of the original categorical distribution. The samples are then further discretized to one-hot vectors using the arg max function. To enable gradient flow during the backward pass, an identity STE replaces the derivative of the arg max function. The GS gradient estimator approach is illustrated in Fig. 1**C**.

Implementing the approach in software

Each model was implemented using the PyTorch library (Stevens et al., 2020) version 2.0.1. PyTorch is a modular Python framework that can be applied to optimize algebraic models by providing automatic differentiation tools (Paszke et al., 2017). The code for all experiments and to reproduce figures is available at `https://surfdrive.surf.nl/files/index.php/s/anTCyB2JV2FVq9X`.

DEFINING SLM CALIBRATION TASKS

Calibration of sLMs to data encompasses various fitting tasks, the applicability of which depends on the type of model, available data, and the target characteristics to emulate. It is thus beneficial if calibration methods apply to a broad range of common tasks. In this work, we consider the following three fitting tasks representing some of the most common applications:

Transition fitting is the task of finding parameters, such that the per-timestep behavior of the model matches a set of observed transitions. This is applicable whenever significant information can be inferred from the transitory behavior of the model. An example of this is found in epidemiology. Considering the per-week spread of Covid within a given region, we can try emulating the spread behavior. It is important to simulate the transitory behavior with high fidelity for the model to be useful in studying the spread of the disease. Considering longer-term trajectories for infectious diseases, on the other hand, does not necessarily lead to an increase in predictability and can even have adverse effects (Scarpino and Petri, 2019).

Trajectory fitting focuses on the longer-term behavior of the model. It involves finding parameters such that certain summary statistics of trajectories generated by the model match a set of target statistics. Being able to perform trajectory fitting is crucial for studying phenomena in non-equilibrium systems. Most commonly, when modelling real-world phenomena, the main interest lies in the long-term behavior of the system which transition fitting does not adequately capture. Due to the stochasticity of the models, reproducing full trajectories is unlikely. The goal, instead, is to match a set of relevant summary statistics to a given target. Example applications of this are systems that exhibit Brownian motion such as stellar bodies (Merritt, 2013) or living cells (Tsekov and Lensen, 2013) where one may aim to match summary statistics such as the mean square displacement (MSD) over time.

Stability fitting is the task of finding parameter configurations for sLMs from which stable patterns arise. For instance, the formation of such stable patterns from an unstable initial configuration over time is an important aspect of biological morphogenesis. Starting with Turing (Turing, 1990), several reaction-diffusion models for the emergence of stable state patterns have been

proposed and studied (Ali and Saleem, 2023; Li et al., 2015; Wang et al., 2011). Independent of the model choice, these stable patterns have been shown to only emerge for a small subset of the parameter space which is hard to identify (Vittadello et al., 2021).

EXPERIMENTS

Transition fitting of a 1-parameter susceptible-infected sLM

Transition fitting is the task of fitting a model to a given set of state transitions – hence, we consider only a single time step of the model. As an example, we use a simple susceptible-infected (SI) sLM that simulates the diffusive spread of a news item through a population. The SI sLM depends on a single "spread coefficient" that determines the rate at which the news spread through the population. In this experiment, we apply AD to recover the spread coefficient from a set of observed transitions.

Model description We consider a spatial variation of a Markov-chain model described in Conlisk (1976). The sLM is defined as a binary 2D square lattice on a periodic torus. Lattice sites in state 1 are considered 'aware' of the news while sites in state 0 are 'unaware'. We denote $L_t^{i,j}$ as the state of the lattice site (i, j) at time t. Further, $N_t^{i,j}$ denotes the number of aware sites in the Moore neighborhood of $L^{i,j}$ at time t. The per-timestep transition likelihood can then be written in a matrix as follows:

$$p(L_{t+1}^{i,j}) = \begin{array}{c|cc|c} & 1 & 0 & L_t^{i,j}/L_{t+1}^{i,j} \\ \hline 1 & 1 & 1-(1-\beta)^{N_t^{i,j}} & 1 \\ 0 & 0 & (1-\beta)^{N_t^{i,j}} & 0 \end{array} \quad (1)$$

Where $p(L_{t+1}^{i,j})$ is a shorthand for the conditional sampling distribution $p(L_{t+1}^{i,j}|L_t^{i,j}, N_t^{i,j}, \beta)$ and β is the spread coefficient. The first column of Eq. 1 encodes that news items cannot be forgotten once learned. The second column models the likelihood of a lattice site turning aware or staying unaware, depending on the number of aware neighbors. At each time step, a new value for each lattice site is simultaneously sampled according to Eq. 1.

The differentiation problem in this case is that the probability in Eq. 1 is discrete and the sampling steps depends on the model parameter β. Hence, we cannot differentiate through the model with respect to β. We apply the GS gradient estimator, as described in Sec. The Gumbel-softmax trick, to circumvent this issue.

Model training We collect four distinct sets of training data by running reference simulations with $\beta = [0.05, 0.2, 0.5, 0.8]$. For each target value, N reference simulations are run and the state transitions (X_t, X_{t+1}) are recorded. The simulations are initialized with a single aware site in the center of the lattice and run for τ steps. For $\beta \leq 0.4$ we use $\tau = 50$ and $N = 100$. For $\beta = 0.8$, we use $\tau = 30$ and $N = 200$ as the boundaries of the grid are reached faster.

The loss function being optimized is the pixel-wise mean-squared error (MSE) between the predicted states $\hat{X}_{t+1} \sim p_{\hat{\beta}}(X_t)$ and observations X_{t+1}. For gradient optimization, minibatch stochastic gradient descent (SGD) (Li et al., 2014) is applied to $\hat{\beta}$ with a batch size of 128 and a learning rate of $1e^{-7}$. The hyperparameters were chosen manually.

Training results Fig. 2 shows the results of fitting the SI sLM to a range of different target values for β. For each target value, a closely matching estimate $\hat{\beta}$ is successfully recovered. This shows that AD via approximate gradients can successfully be used to perform transition fitting for sLMs.

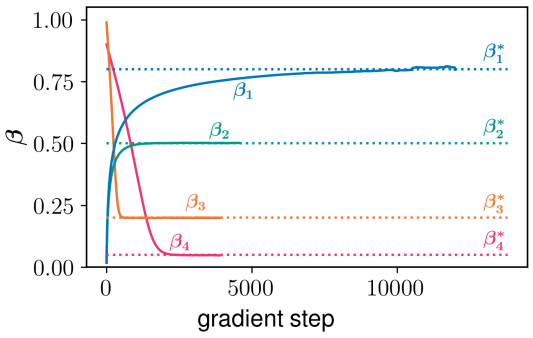

Figure 2: Transition fitting the spread coefficient of a simple SI sLM via AD can recover a close estimate of the original β for a range of target values.
We fit β to four different target values $\beta_1^* = 0.8, \beta_2^* = 0.5, \beta_3^* = 0.2, \beta_4^* = 0.05$. The plot shows the parameter trace for each β_i during gradient optimization.

Standing out from the results is that fitting β converges significantly slower for higher values of β compared to lower values. The reason for this is that for higher spread coefficients the rate of spread saturates and trajectories become very similar to another.

Trajectory fitting of a persistent random walk sLM

While transition fitting is a straightforward task that is simple to define, in many sLM applications we will be more interested in capturing the longer-term behaviour of a system. A natural generalization of transition fitting is to consider longer chains of transitions *trajectories*. Due to the stochasticity of sLMs, our goal is not to exactly reproduce a given set of trajectories; instead, we seek to match some given set of summary statistics between our model and the data.

As an example, we consider a persistent random walk sLM. For sufficiently many steps, it is unlikely to reproduce any given random walk sequence. Instead, random walk models are typically fitted to summary statistics such as the MSD over time of a given set of trajectories. Here, we consider the movement of different types of cells that are known to exhibit a persistent random walk over time. Specifically, we apply AD via

approximate gradients to fit the parameters of a persistent random walk sLM to the MSD collected in Wortel et al. (2021a) for T cells, B cells, and Neutrophils.

Model description We model the persistent random walk of an agent on a binary regular square lattice as two independent random walks along the x and y axes. At any given time t the agent is defined by the pixel (i, j) it occupies and the current velocity $v_x, v_y \in \{-1, 0, 1\}$. We define one Monte Carlo step (MCS) as the combination of taking a step in both x and y with the respective velocity.

At every MCS, we retain the velocity of the previous step with a probability $1 - p_{\text{resample}}$. Thus with a probability of p_{resample}, we resample the velocities according to a distribution:

$$v = \begin{cases} -1, & \text{with } p = {}^{(1\,-\,p_{\text{center}})}/_{2} \\ 0, & \text{with } p = p_{\text{center}} \\ 1, & \text{with } p = {}^{(1\,-\,p_{\text{center}})}/_{2} \end{cases}$$

Both p_{center} and p_{resample} are logit-transformed during fitting for improved numerical stability.

The model contains two discrete sampling operations depending on the model parameters. First, the binary decision of resampling or keeping the current velocity depends on p_{resample}. Second, sampling a new velocity from the categorical distribution over $\{-1, 0, 1\}$ depends on p_{center}. To make the model differentiable, we approximate the gradient of the sampling operations using the GS gradient estimator as shown in Fig. 1C and described in Sec. The Gumbel-softmax trick.

Model training For each step of gradient descent (GD), we simulate a batch of 1,500 lattices for 10 steps to match the number of datapoints. As a loss function, we calculate the sum of mean squared errors (MSEs) between the MSD of the simulated batch and the target MSD for every datapoint and take the sum of the errors. We apply the Adam optimization scheme (Kingma and Ba, 2014) with the standard parameters of PyTorch and a manually chosen learning rate of 0.01.

For the cell data we consider a dataset of trajectories where the MSD is tracked periodically every 24 seconds over a certain time frame of which we consider the first ten datapoints. In the sLM of the cell data, we define $1\text{MCS} = 24s$ as well as $\Delta x = \sqrt{10}\mu m$. Limiting the datasets to a comparatively small amount of 10 datapoints pronounces the persistence effects, which vanish for larger time frames Fürth (1920). In addition to the experimental data, we synthetically generate an MSD curve with our model for known parameters $p_{\text{center}} = 0.2, p_{\text{resample}} = 0.1$ by simulating a batch of 15,000 trajectories over 10MCS each.

Training results Fig. 3A shows the results of recovering the target parameters from synthetically generated data. Starting from bad initial guesses, the estimated parameters converge to closely match the target parameters after about 5,000 steps of gradient descent.

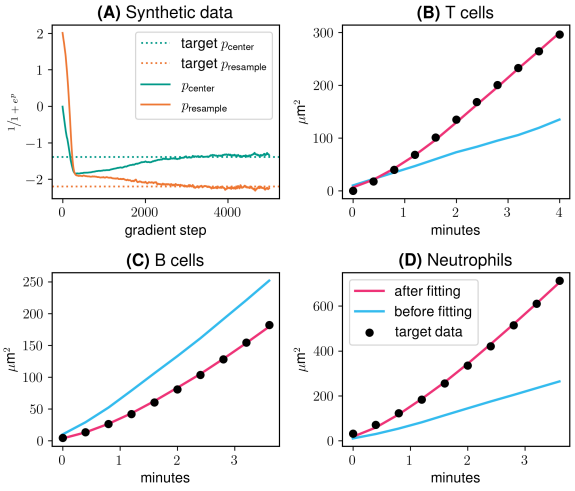

Figure 3: AD via approximate gradients can successfully perform trajectory fitting on a persistent random walk sLM to cell movement as well as recover parameters. **(A)** AD via approximate gradients can recover parameters by applying the method to synthetic data. **(B),(C),(D)** We apply trajectory fitting using the MSD curve of (A) a set of T cells, (B) a set of B cells (C) a set of Neutrophils over time.

Fig. 3B shows the results of fitting the sLM to a dataset of T cell movement. Persistent brownian motion exhibits an initial non-linear 'onramping' phase that transitions into quasi-linear growth over time. The steepness of the onramping is mainly governed by the resampling probability p_{resample} whereas the transition time between onramping and linear growth is governed by the probability of moving p_{center}. After 10,000 steps of Adam optimization, the MSD curve generated by the optimized model closely matches the MSD curve of the T cell dataset.

Fig. 3C,D show similar results for the experiments using B cell data and Neutrophil data, respectively. In each case, the data follows the persistent random walk pattern of non-linear onramping transitioning into quasi-linear growth of the MSD. We start from bad initial guesses for the parameters, which produce MSD curves that match neither the persistence pattern nor the average displacement over time of the data. After 10,000 steps of Adam optimization in both cases, we attain parameter estimates that produce MSD curves closely matching those of the target data.

The results show that starting from bad initial parameters, we can successfully make use of AD via approximate gradients to estimate a set of parameters $p_{\text{center}}, p_{\text{resample}}$ for which the MSD curve closely matches that of the target data. Hence, the results suggest that AD via approximate gradients can be used for trajectory fitting sLMs to data.

Stability fitting of a reaction-diffusion sLM for pattern generation

Finally, we investigate applying stability fitting to a reaction-diffusion sLM that produces stable patterns for certain regions of the parameter space. Being able

to apply AD for this type of fitting would provide an efficient way of locating the subspaces that generate Turing patterns. Further, the training process can provide insights into the model, highlighting the impact of the different parameters on pattern formation.

Model description We consider the "Malevanets-Kapral" reaction-diffusion sLM introduced in Malevanets and Kapral (1997). It is a spatial reformulation of the Fitzhugh-Nagumo equations (FitzHugh, 1961), which simulates the reaction-diffusion of two chemical species over time. For an in-depth description of the model we refer to Malevanets and Kapral (1997). Generally, the Malevanets-Kapral sLM models per-chemical concentration of two chemical species A, B in space as a 2-layer square lattice $L = (A, B)$. $A^{i,j}$ represents the concentration of species A at the lattice site (i, j), whereas $B^{i,j}$ denotes the concentration of species B at the same lattice site. Every site has a maximum capacity of N molecules per species. At every time step, the model simulates:

1. Independent diffusion of A and B, governed by the diffusion coefficients D_A, D_B
2. The reaction of A and B with each other, using mass-action kinetics with reaction rates k_1, k_2, k_3

The non-differentiability of this model stems from two sources. First, reactions happen with a probability depending on the reaction rate of the corresponding reaction channel. We approximate the gradient by applying reparameterization in combination with a STE. Second, diffusion involves random direction sampling followed by translating the grid in the chosen direction. To approximate the gradient of the categorical sampling, we apply a GS gradient estimator.

Additionally, a parameter γ scales the rate of the reaction process to the diffusion process. In our experiments, we fixed $\gamma = 0.005$. By iteratively applying the two steps to the lattice, the reaction-diffusion process of the two chemicals is simulated. For a subset of parameters, this interaction generates stable patterns, such as shown in Fig. 4. Here, we test:

E1 : Joint optimization of the model parameters $\Theta = (D_A, D_B, k_1, k_2, k_3)$

E2 : Optimization of the reaction rates k_1, k_2, k_3 for fixed diffusion coefficients

E3 : Optimization of the diffusion coefficients D_A, D_B for fixed reaction rates

We investigate whether reaction or diffusion rates are more difficult to estimate, and whether joint optimization of both is feasible.

Model training Pattern formation performance is challenging to accurately capture in a loss function (Vittadello et al., 2021). Inspired by the training strategy used by Mordvintsev et al. (2020) for optimizing continuous "Neural Cellular Automata" models towards generating stable patterns, we use pattern maintenance as an alternative measure that is easily quantifiable. Thus for each gradient step, the evolution of a given stable pattern X_{ref} is simulated for τ timesteps to produce $X_{ref+\tau}$. The pixel-wise MSE of $(X_{ref}, X_{ref+\tau})$ then serves as a surrogate loss function. To validate how well this generalizes to pattern formation, we perform *pattern formation tests* during optimization using the current parameters. In each test, we simulate a system starting from a uniform, unstable state and observe if stable patterns emerge.

Each experiment is performed on 64×64 square lattices on a periodic torus. For optimization, standard gradient descent is applied with a learning rate of 0.05. The learning rate was again chosen by hand through experimentation. Each simulation is run for $\tau = 500$ steps with a reaction timescale $\gamma = 0.005$ and maximum capacity $N = 50$. For experiment 2, we fixed $k_1 = 0.98$, $k_2 = 0.1$ and $k_3 = 0.2$, based on results reported in Malevanets and Kapral (1997). Similarly, for experiment 3 we fixed $D_A = 0.1$ and $D_B = 0.4$.

Training results For experiment *E1*, 6,000 steps of gradient descent are shown in Fig. 4A. For the initial parameters, no patterns are formed. After 3,000 steps, the test sample shows an overall higher concentration of species A with a small pattern in the spatial distribution. Finally, the test sample after 6,000 steps shows clear patterns in the spatial distribution of species A. Thus, the gradient-based fitting method has successfully identified a set of morphogenetic parameters.

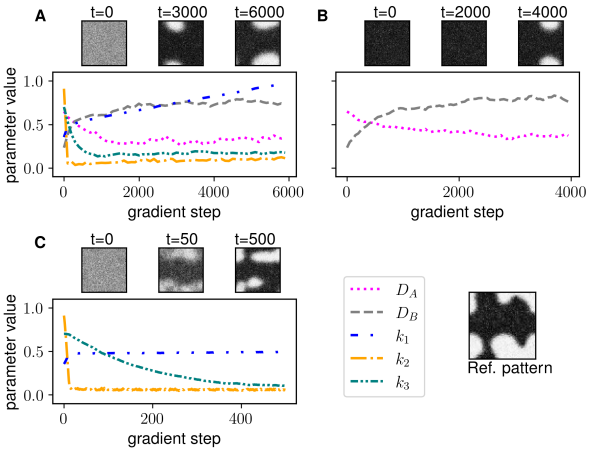

Figure 4: Morphogenetic parameter configurations of the Malevanets-Kapral model are found through gradient-based stability fitting using AD.
A Joint fitting of the reaction rates and diffusion coefficients. **B** Fitting of the diffusion coefficients for fixed reaction rates. **C** Fitting of the reaction rates for fixed diffusion coefficients. **Top Row** Samples of the pattern formation test. Depicted: concentration of species A after simulation. **Bottom Row** Parameter trace of the fitting process.

The results of experiment *E2* are summarized in Fig. 4B. Compared to *E1*, the initial parameter configuration does not produce a white noise pattern. Instead, the concentration of species A appears to have con-

verged to maximum capacity for all lattice sites. This state is stable, but no patterns are formed. After 4,000 steps the test sample shows a clear stable pattern in the spatial distribution of species A.

The diffusion coefficients in E2 start with $D_A > D_B$. After 500 gradient steps this relation is reversed and after 4000 steps D_B is roughly twice as large as D_A. Malevanets and Kapral (1997) note in their analysis of the system that the ratio D_B/D_A needs to exceed a critical value for labyrinthine stable patterns to form, which is mirrored by the parameter trace.

Finally, the results of experiment *E3* are shown in Fig. 4C. Fitting the reaction rates takes significantly fewer gradient steps than the diffusion coefficients. After only 50 steps of gradient descent, the test sample shows the first patterns, albeit not appearing fully stable. Then after 500 steps, clear stable patterns are formed. Comparing the results of experiments *E2* and *E3*, fitting the reaction rates appears to be significantly easier compared to fitting the diffusion coefficients.

Overall, the results of the experiments show that it is possible to apply AD via approximate gradients to identify regions within the parameter space of growth sLMs that generate stable patterns. This was shown for two types of parameters; diffusion coefficients and reaction rates. While the results suggest that fitting diffusion coefficients is more challenging, the method was successful in both cases.

DISCUSSION

We investigated an alternative approach for gradient-based optimization of sLMs, using AD via approximate gradients. To this end, three common types of fitting tasks pertaining to sLM were identified. For each type, a series of experiments was performed to investigate the viability of the proposed method.

Taken together, our experimental results show preliminary evidence that AD via approximate gradients is a viable and versatile strategy to fit sLM parameters to data. As a proof of concept, we apply AD via approximate gradients to three simple models taken from different domains of research that require different fitting tasks. Further, we provide an easy-to-use codebase that can be referenced for further experiments.

However, there are some drawbacks and possible limitations to consider for the gradient-based approach presented here. Foremost it has to be acknowledged that gradient-free methods such as ABC can perform black-box optimization of existing model implementations. This is not the case for the gradient-based approach as we need to re-implement the models to enable differentiability. Further, while gradient descent is often effective it introduces new hyperparameters. The learning rate, for example, can have a significant impact on convergence (Jacobs, 1988). More sophisticated gradient descent algorithms introduce additional hyperparameters (Kingma and Ba, 2014). For large neural networks with millions of parameters, these additional hyperparameters are a small price to pay for efficient optimization. sLMs, on the other hand have relatively few parameters and the number of hyperparameters could easily be larger. In this scenario, we have simply replaced the original estimation problem with another one of possibly the same complexity. Further challenges of gradient-based optimization methods to consider are convergence to local optima and the problem of defining the right cost function.

In future research, we aim to apply the method to more complex sLMs such as the Cellular Potts Model for cells and tissues (Graner and Glazier, 1992). Additionally, it is of interest to investigate the possibility of combining multiple fitting tasks into a joint optimization scheme. The performance of the method should be compared to parameter calibration via gradient-free calibration methods such as ABC. While these methods are not directly comparable in that they do not solve the same task (posterior inference compared to mode estimation), we could use for example the wall clock time of the methods to establish a comparison of efficacy for calibration.

ACKNOWLEDGEMENTS

Part of the research presented was performed for an ELLIS excellence fellowship, funded by Radboud AI, Radboud University. This research was further supported by HFSP program grant RGP0053/2020.

References

E. Alamoudi, Y. Schälte, R. Müller, J. Starruss, N. Bundgaard, F. Graw, L. Brusch, and J. Hasenauer. Fitmulticell: Simulating and parameterizing computational models of multi-scale and multi-cellular processes. *bioRxiv*, pages 2023–02, 2023.

I. Ali and M. T. Saleem. Spatiotemporal dynamics of reaction–diffusion system and its application to turing pattern formation in a gray–scott model. *Mathematics*, 11(6):1459, 2023.

P. Andelfinger. Towards differentiable agent-based simulation. *ACM Transactions on Modeling and Computer Simulation*, 32(4):1–26, 2023.

G. Arya, M. Schauer, F. Schäfer, and C. Rackauckas. Automatic differentiation of programs with discrete randomness. *Advances in Neural Information Processing Systems*, 35:10435–10447, 2022.

M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4): 2025–2035, 2002.

Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

H.-G. Beyer, M. Olhofer, and B. Sendhoff. On the behavior of (μ/μá, λ)-es optimizing functions disturbed by generalized noise. *Foundations of Genetic Algorithms*, 7, 2002.

P. Cannon, D. Ward, and S. M. Schmon. Investigating the impact of model misspecification in neural simulation-based inference. *arXiv preprint arXiv:2209.01845*, 2022.

M. J. Carr, M. J. Simpson, and C. Drovandi. Estimating parameters of a stochastic cell invasion model with fluorescent cell cycle labelling using approximate bayesian computation. *Journal of the Royal Society Interface*, 18(182):20210362, 2021.

B. Chopard and M. Droz. "cellular automata modeling of physical systems", 2005, 2005.

A. Chopra, A. Rodríguez, J. Subramanian, A. Quera-Bofarull, B. Krishnamurthy, B. A. Prakash, and R. Raskar. Differentiable agent-based epidemiology. *arXiv preprint arXiv:2207.09714*, 2022.

J. Conlisk. Interactive markov chains. *Journal of Mathematical Sociology*, 4(2):157–185, 1976.

D. Dab, J.-P. Boon, and Y.-X. Li. Lattice-gas automata for coupled reaction-diffusion equations. *Physical review letters*, 66(19): 2535, 1991.

K. Durso-Cain, P. Kumberger, Y. Schälte, T. Fink, H. Dahari, J. Hasenauer, S. L. Uprichard, and F. Graw. Hcv spread kinetics reveal varying contributions of transmission modes to infection dynamics. *Viruses*, 13(7):1308, 2021.

J. Dyer, P. Cannon, J. D. Farmer, and S. Schmon. Black-box bayesian inference for economic agent-based models. *arXiv preprint arXiv:2202.00625*, 2022.

R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466, 1961.

D. T. Frazier, C. P. Robert, and J. Rousseau. Model misspecification in approximate bayesian computation: consequences and diagnostics. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(2):421–444, 2020.

R. Fürth. Die brownsche bewegung bei berücksichtigung einer persistenz der bewegungsrichtung. mit anwendungen auf die bewegung lebender infusorien. *Zeitschrift für Physik*, 2(3):244–256, 1920.

F. Graner and J. A. Glazier. Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical review letters*, 69(13):2013, 1992.

N. Hansen, A. Ostermeier, and A. Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In *ICGA*, pages 57–64, 1995.

C. A. Haselwandter and D. D. Vvedensky. Renormalization of stochastic lattice models: Epitaxial surfaces. *Physical Review E*, 77(6):061129, 2008.

I. A. Huijben, W. Kool, M. B. Paulus, and R. J. Van Sloun. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1353–1371, 2022.

R. A. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4):295–307, 1988.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

R. P. Kelly, D. J. Nott, D. T. Frazier, D. J. Warne, and C. Drovandi. Misspecification-robust sequential neural likelihood. *arXiv preprint arXiv:2301.13368*, 2023.

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

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

M. Li, T. Zhang, Y. Chen, and A. J. Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 661–670, 2014.

S. Li, J. Wu, and Y. Dong. Turing patterns in a reaction–diffusion model with the degn–harrison reaction scheme. *Journal of Differential Equations*, 259(5):1990–2029, 2015.

J.-M. Lueckmann, J. Boelts, D. Greenberg, P. Goncalves, and J. Macke. Benchmarking simulation-based inference. In *International conference on artificial intelligence and statistics*, pages 343–351. PMLR, 2021.

A. Malevanets and R. Kapral. Microscopic model for fitzhugh-nagumo dynamics. *Physical review E*, 55(5):5657, 1997.

K. L. Mengersen, P. Pudlo, and C. P. Robert. Bayesian computation via empirical likelihood. *Proceedings of the National Academy of Sciences*, 110(4):1321–1326, 2013.

D. Merritt. *Dynamics and evolution of galactic nuclei*. Princeton University Press, 2013.

A. Mordvintsev, E. Randazzo, E. Niklasson, and M. Levin. Growing neural cellular automata. *Distill*, 5(2):e23, 2020.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

A. Quera-Bofarull, A. Chopra, J. Aylett-Bullock, C. Cuesta-Lazaro, A. Calinescu, R. Raskar, and M. Wooldridge. Don't simulate twice: One-shot sensitivity analyses via automatic differentiation. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1867–1876, 2023a.

A. Quera-Bofarull, J. Dyer, A. Calinescu, and M. Wooldridge. Some challenges of calibrating differentiable agent-based models. *arXiv preprint arXiv:2307.01085*, 2023b.

S. V. Scarpino and G. Petri. On the predictability of infectious disease outbreaks. *Nature communications*, 10(1):898, 2019.

H.-J. M. Shi, M. Q. Xuan, F. Oztoprak, and J. Nocedal. On the numerical performance of derivative-free optimization methods based on finite-difference approximations. *arXiv preprint arXiv:2102.09762*, 2021.

S. A. Sisson, Y. Fan, and M. Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

E. Stevens, L. Antiga, and T. Viehmann. *Deep learning with PyTorch*. Manning Publications, 2020.

H. Sumata, F. Kauker, R. Gerdes, C. Koeberle, and M. Karcher. A comparison between gradient descent and stochastic approaches for parameter optimization of a sea ice model. *Ocean Science*, 9 (4):609–630, 2013.

A. Szabó and R. M. Merks. Cellular potts modeling of tumor growth, tumor invasion, and tumor evolution. *Frontiers in oncology*, 3: 87, 2013.

J. Tang, J.-H. Lai, W.-S. Zheng, L. Yang, and X. Xie. Relaxation lif: A gradient-based spiking neuron for direct training deep spiking neural networks. *Neurocomputing*, 501:499–513, 2022.

R. Tsekov and M. C. Lensen. Brownian motion and the temperament of living cells. *Chinese Physics Letters*, 30(7):070501, 2013.

A. M. Turing. The chemical basis of morphogenesis. *Bulletin of mathematical biology*, 52:153–197, 1990.

S. T. Vittadello, T. Leyshon, D. Schnoerr, and M. P. Stumpf. Turing pattern design principles and their robustness. *Philosophical Transactions of the Royal Society A*, 379(2213):20200272, 2021.

W. Wang, Y. Lin, F. Yang, L. Zhang, and Y. Tan. Numerical study of pattern formation in an extended gray–scott model. *Communications in Nonlinear Science and Numerical Simulation*, 16(4): 2016–2026, 2011.

X. Wang, A. L. Jenner, R. Salomone, D. J. Warne, and C. Drovandi. Calibration of agent based models for monophasic and biphasic tumour growth using approximate bayesian computation. *bioRxiv*, pages 2022–09, 2022.

D. Ward, P. Cannon, M. Beaumont, M. Fasiolo, and S. Schmon. Robust neural posterior estimation and statistical model criticism. *Advances in Neural Information Processing Systems*, 35:33845–33859, 2022.

I. M. Wortel, A. Y. Liu, K. Dannenberg, J. C. Berry, M. J. Miller, and J. Textor. Celltrackr: an r package for fast and flexible analysis of immune cell migration data. *ImmunoInformatics*, 1:100003, 2021a.

I. M. Wortel, I. Niculescu, P. M. Kolijn, N. S. Gov, R. J. de Boer, and J. Textor. Local actin dynamics couple speed and persistence in a cellular potts model of cell migration. *Biophysical journal*, 120(13):2609–2622, 2021b.

AUTHOR BIOGRAPHIES

JAN D. SCHERING is a doctoral candidate in the cross-section of Bioinformatics and Artificial Intelligence at the Radboud University in Nijmegen, the Netherlands. He is the main correspondent for questions relating to this work. His email address is: `jan.schering@ru.nl`

SANDER KEEMINK is an assistant professor in computional neuroscience and AI at the Donders Institute for Brain, Cognition and Behavior at the Radboud University in Nijmegen, The Netherlands. His main research focus is to find explanations for network function in artificial and biological neural networks.

JOHANNES TEXTOR works both at the Radboud University and the Radboud University Medical Center in Nijmegen, The Netherlands. He is interested in leveraging causal inference methodology for the benefit of biomedical research, especially in the fields of Immunology and Tumor Immunology. He also works on the development of causal inference methodology and on simulation and modeling more generally.