Towards Modeling Geographical Processes with Generative Adversarial Networks (GANs)

David Jonietz[1]
HERE Technologies, Switzerland
david.jonietz@here.com

Michael Kopp
HERE Technologies, Switzerland
Institute of Advanced Research in Artificial Intelligence (IARAI), Austria
michael.kopp@here.com

Abstract

Recently, Generative Adversarial Networks (GANs) have demonstrated great potential for a range of Machine Learning tasks, including synthetic video generation, but have so far not been applied to the domain of modeling geographical processes. In this study, we align these two problems and – motivated by the potential advantages of GANs compared to traditional geosimulation methods – test the capability of GANs to learn a set of underlying rules which determine a geographical process. For this purpose, we turn to Conway's well-known Game of Life (GoL) as a source for spatio-temporal training data, and further argue for its (and simple variants of it) usefulness as a potential standard training data set for benchmarking generative geographical process models.

2012 ACM Subject Classification Computing methodologies → Neural networks

Keywords and phrases GAN, generative modeling, deep learning, geosimulation, game of life

Digital Object Identifier 10.4230/LIPIcs.COSIT.2019.27

Category Short Paper

1 Introduction

Contrary to the inherently rather space-focused perspective of Geographical Information Systems (GIS), spatial systems are in general highly dynamic. Thus, the involved geographical entities are susceptible to change with regards to their spatial (e.g., appearance, disappearance, expansion, contraction, movement) or thematic domain (changes of one or more attributes) [6, 4]. Modeling such dynamic behavior is of critical importance for a wide range of applications (e.g., transport planning and traffic prediction, weather forecasting, or disaster management), and involve both explanatory and predictive modeling approaches [20].

Explanatory models are typically targeted towards reaching a thorough understanding of the modeled domain. Relevant features are described in the form of a causal theoretical model, which is then either tested statistically or by hand-crafting a set of fundamental behavioral rules, running simulations and exploring different scenarios (e.g., traffic demand modeling, wildfire spread simulation). In geosimulation applications, in contrast to more aggregate statistical modeling approaches, the elementary system units are typically modeled in great detail as individual automata, using paradigms such as Cellular Automata (CA) or Agent-based Models (ABM) [3].

[1] corresponding author

14th International Conference on Spatial Information Theory (COSIT 2019).
Editors: Sabine Timpf, Christoph Schlieder, Markus Kattenbeck, Bernd Ludwig, and Kathleen Stewart; Article No. 27; pp. 27:1–27:9
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Nowadays, however, due to an increased usage of geo-sensors, large-scale spatio-temporal data sets are widely available which describe various geographical processes with an unprecedented level of detail. Thus, for predictive models, where the focus is put less on understanding the functional principles of the system but rather on predicting its next state based on previous observations [20], supervised machine learning algorithms such as Artificial Neural Networks (ANN) have been increasingly used, e.g., for short-term traffic forecasting [14, 16] or precipitation prediction [26]. In contrast to explanatory approaches, these models are based on data rather than theory, and can usually be trained in an end-to-end manner. Abstract features of relevance for the predictive task are learned directly from the data, however, to what degree such models reach a true understanding of the problem domain remains largely unclear.

Recently, a new wave of generative models has had a disruptive effect on the Machine Learning community, which aim to generate realistic samples of a complex, real world distribution having only observed true samples of said distribution. Thus, being presented with data (e.g., images, text, music or videos), these models move beyond predictive models by learning a representation which particularly encodes important semantic features in order to generate new, hitherto unseen, 'realistic' samples, therefore potentially understanding the underlying data-generating process itself [5]. A particularly successful example for these models are Generative Adversarial Networks (GANs) [8], which require no prior assumptions or hypotheses about the function principles of the modeled system.

To the best of our knowledge, our work represents the first study which explores the potential of GANs for the simulation of geographical processes. This is motivated by the fact that in our view, GANs combine strengths of both explanatory and predictive modeling approaches. GANs, as used here, are explanatory with regards to a geographical process as they capture its underlying hidden rules and -on this basis- are able to not only generate novel sample states, but also provide a learned loss metric which describes how "realistic" a given sample is. In contrast to traditional explanatory models, however, GANs do not rely on hand-crafted parameters (as in expert systems), but directly learn them from observation alone while preserving the capability of capturing and applying complex rules (one could thus refer to them as "self-learned explanatory" models). If the generated samples describe future states of a process a GAN can be used as predictive model, thereby eliminating the need for descriptive rules or a set framework (cf. variational bayesian methods or deterministic methods such as SVMs) while preserving the apparent ability to sample highly complex naturally occurring distributions.

At this stage, we aim to demonstrate and quantify the performance of a GAN on a well-controlled test data set (as it is the only way of measuring the effectiveness of most other neural network architectures as well). For this, we choose a straight-forward example of a complex, non-trivial and non-deteriorating geo-spatial process that arises out of a simple set of deterministic rules: Conway's Game of Life (GoL) [7]. In general, in view of the multitude and diversity of potential use cases from different geo-spatial domains (and the according spatio-temporal data sets), we argue for the general need for a standard training data set for benchmarking generative geographical process models (comparable to MNIST [12] for image processing tasks), and propose to use the GoL - and selected adaptations - for this purpose. In our experiments, we demonstrate that a GAN can indeed learn the underlying rules of the data-generating process (and therefore play the GoL correctly), however, processes with different properties require different network architectures.

2 Generative Adversarial Networks (GANs)

GANs aim to capture the statistical distribution of training data and produce new, hitherto unseen, samples from that distribution. In its original form [8] each GAN model has two parts to it that compete against each other: a generator whose task it is to produce new "fake" samples from the underlying distribution of the observed data (forger) and a discriminator who, when faced with "real" and "fake" samples aims to tell them apart (policeman). The Generator G and Discriminator D can be defined as a functions

$$G_{\theta_g} : \{\text{random input}\} \longrightarrow \{\text{Samples}\}, z \longrightarrow x \tag{1}$$

$$D_{\theta_d} : \{\text{Samples}\} \longrightarrow [0,1], x \longrightarrow k \tag{2}$$

where z refers to an input random noise variable, which is mapped via G to a sample x in data space based on a set of parameters θ_g. Its counterpart D represents a function where any input sample x is mapped to a scalar k which expresses the probability that x is sampled from the original statistical distribution rather than created by G, based on a set of parameters θ_d. Typically, both functions G and D would be implemented as separate neural networks.

The training process outlined in [8] is defined by D and G playing a two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_r}[logD(x)] + \mathbb{E}_{z \sim p_z}[log(1 - D(G(z)))] \tag{3}$$

where p_r is the data distribution (in many cases unknowable) from which 'real' samples x_r are drawn and p_z is the data distribution over noise input z. Although GANs have been able to generate photo-realistic images, there is currently no known way of quantifying how well the generator in general approximates the original distribution. In particular, in some cases, GANs are known to experience mode collapse and a plethora of techniques are employed to mitigate this phenomena (see for instance [19]). So far, GANs have so far been rarely used in the geospatial domain, which is mainly due to their relative novelty and notoriety to be difficult to train. Exemplary applications have been set mostly in a remote sensing context (e.g., [27]), but also included e.g., the generation of static traffic [16] or urbanization patterns [1].

In this study, we use a conditional GAN [18], an extended concept where a conditional input value y is added to the random input z in the formula above so that the aim of the GAN is to produce samples from the corresponding conditional distribution:

$$G_{\theta_g} : \left\{ \text{ random input, conditional input } \right\} \longrightarrow \{\text{samples}\}, z, y \longrightarrow x \tag{4}$$

$$D_{\theta_d} : \{\text{samples , conditional input}\} \longrightarrow [0,1], x, y \longrightarrow k \tag{5}$$

In the past, GANs conditioned with past frames of videos have been successfully applied to next frame prediction tasks (see e.g., [15, 13]). In [10], a conditional GAN was used for augmenting the training set for a traffic prediction task. In this work, by modeling a geographical process using the traditional snapshot approach [2], where each 'frame' depicts a time-stamped map view of the current state of the spatial system, we conceptually align the tasks of spatio-temporal modeling and synthetic video generation.

3 Conway's Game of Life

Conway's Game of Life is a popular example of a CA-based game (see [7]). Formally, a CA can be defined as a discrete dynamic system consisting of a n-dimensional fixed lattice arrangement of cells C, each cell $c \in C$ being in a certain state $s_c(t) \in S$ during a discrete time step t where the value lies in some set S. We shall restrict our attention to $S = \{0, 1\}$. At time step $t + 1$, it is succeeded by a state which can be described by a transition function φ taking only into account the previous state $s_c(t)$ as well as the previous state of neighbours of all direct neighbours of c in the lattice C. We shall restrict our attention to games where the dependence on the neighbours is indirect, given by a function on the neighbouring states, such as the sum of all 1s occurring. Formally, if N_c denotes the set of neighbours of c in the lattice C and $f_{N_c}(t)$ denotes the function value at time step t for the neighbors of c, then

$$s_c(t+1) = \varphi\left(s_c(t), f_{N_c}(t)\right) \tag{6}$$

Moving the perspective of the state of an individual cell $c \in C$ to all states configurations of C at a given time step t, we define

$$X(t) = \{s_c(t)\,|c \in C\} \tag{7}$$

as the configuration of a CA at time t.

Conway's Game of Life is set on the two-dimensional square lattice $\mathbb{Z}^2$ (where each cell has precisely 8 neighbours) with only two states for each cell and simple rules given, with the notation above, by

$$\varphi(0,x) = \begin{cases} 0 & \text{if } x \neq 3 \\ 1 & \text{if } x = 3 \end{cases} \qquad \varphi(1,x) = \begin{cases} 0 & \text{if } x < 2 \text{ or } x > 3 \\ 1 & \text{if } x \in \{2,3\} \end{cases} \tag{8}$$

If one calls cells with a value 1 and 0 "alive" and "dead", respectively, one can interpret this update rule in terms of survival (cells with 2 or 3 "alive" neighbours stay "alive"), death (through overpopulation or isolation) and birth, see [7]. In order to simulate the game on finite computer architectures, most implementations restrict their view of the lattice $\mathbb{Z}^2$ to $\{0, 1, \cdots, N-1\}^2$ and decreeing that the state value of neighbours on the boundary of that lattice point square, but outside of it, have state zero.

Despite its simplicity, this game exhibits a surprising variety of oscillating, population increasing and self-replicating state patterns ([7], [22], [25]). In our view, it also represents a powerful abstraction of geographical processes in general, and is therefore a well-suited case study for benchmarking models. Thus, the GoL exhibits similarities to well-known attributes of geographical processes such as the conceptualization of objects, states, processes and events [21], properties related to process dynamics like initiation, cessation and constancy [4], or systemic attributes such as location, topology, spatial interaction [11], or emergence [3].

4 Method

In our experiments, we aim to test whether a GAN can learn the underlying rules of a geographical process, at this stage abstracted as a GoL simulation. For this, we train a GAN on the task of playing the game, i.e., generating the correct next cell configuration $\hat{X}(t)$ while being conditioned on the previous $X(t-n:t)$ configurations (here $n = 4$). Both the generator and the discriminator, therefore, have to internalize the game's transition rules, state space and neighborhood definition in order to successfully fool or expose their counterpart.

4.1 Adaptations of the Game of Life

As discussed previously, numerous properties of the GoL qualify it as a useful abstraction from real-world geographical processes, such as spatial and temporal locality (Moore neighborhood & Markov property) and spatio-temporal dependence. It is clear, however, that most real-world processes are guided by much more complex rules, which, as we argue, can be approximated by manipulating one or more parameters of the traditional game definition. Thus, for instance, more complex spatio-temporal dependencies could be achieved by abolishing the Markov property and introducing more complex, non-uniform neighborhood definitions. Other possible adaptations could include replacing the traditional deterministic with stochastic transition rules, among others.

As a first example for such adaptations, we test our GAN on two versions of the GoL, one following the traditional game definition (in the following: GoL I) and an adapted one (GoL II), where the neighborhood concept is re-defined as follows: If the cell at lattice point $(i, j) \in \{0, 1, \cdots, N-1\}^2$ is denoted by $c(i, j)$, then, if precisely one of $i < \frac{N}{2}$ and $j < \frac{N}{2}$ holds, we replace the neighbourhood $N_{c(i,j)}$ with

$$N^*_{c(i,j)} = \left\{ c(l, k) \,\middle|\, c(k, l) \in N_{c(i,j)} \right\} \tag{9}$$

It is not hard to see that, as sets, $N^*_{c(i,j)} = N_{c(j,i)}$ and since φ is only dependent on the sum of state values over those sets, the upshot of these operations is that we replaced neighbourhoods in the top right quadrant with those corresponding ones in the bottom left through transposition (if we stipulate that i and j, as is the case in matrices, grow from left to right and top to bottom, respectively, in order to define the quadrants above). Thus, with GoL II, the conditions of spatial proximity and homogeneity for defining the neighborhood of cells are dismissed.

From both GoL I and II, we sample 30 000 frame sequences of length 5 frames, each randomly initialized, and split them into training (90%) and test set (10%). For each of the samples in both sets, we use the first 4 frames as conditional input for both the generator and the discriminator, and generate - or discriminate, respectively- the subsequent final frame.

4.2 GAN Architecture

Our GAN architecture (see figure 1) is based on the convolutional long short term memory (convLSTM) approach which has proven successful for a similar spatio-temporal prediction task [26]. Concretely, in the generator the conditional input is encoded via three convLSTM layers with 128 (3×3), 128 (3×3), and 1 (3×3) filters with stride 1, and concatenated with the noise vector z, which has previously been encoded via 2 dense layers with 400 units and leaky ReLU activations [17]. Finally, the encoded features flow through two additional dense layers with each 400 units and leaky ReLU and a final sigmoid activation.

In the discriminator, the conditional input and the predicted $\hat{X}(t)$ or real frame $X(t)$ are concatenated and encoded by two convLSTM layers with 64 (3×3), and 64 (3×3) filters and stride 1. Instead of a noise vector, however, the encoded features are concatenated with the output of a minibatch discrimination layer [19] in order to prevent mode collapse, and then fed through a dense layer with 32 units and leaky ReLU before a final 1 unit sigmoid activation. To prevent the discriminator from completely dominating the generator, we apply drop-out with a rate of 0.6 to the former, thus randomly dropping 60 % of units during each training batch of both the last convLSTM layer and the first dense layer (see [23]).

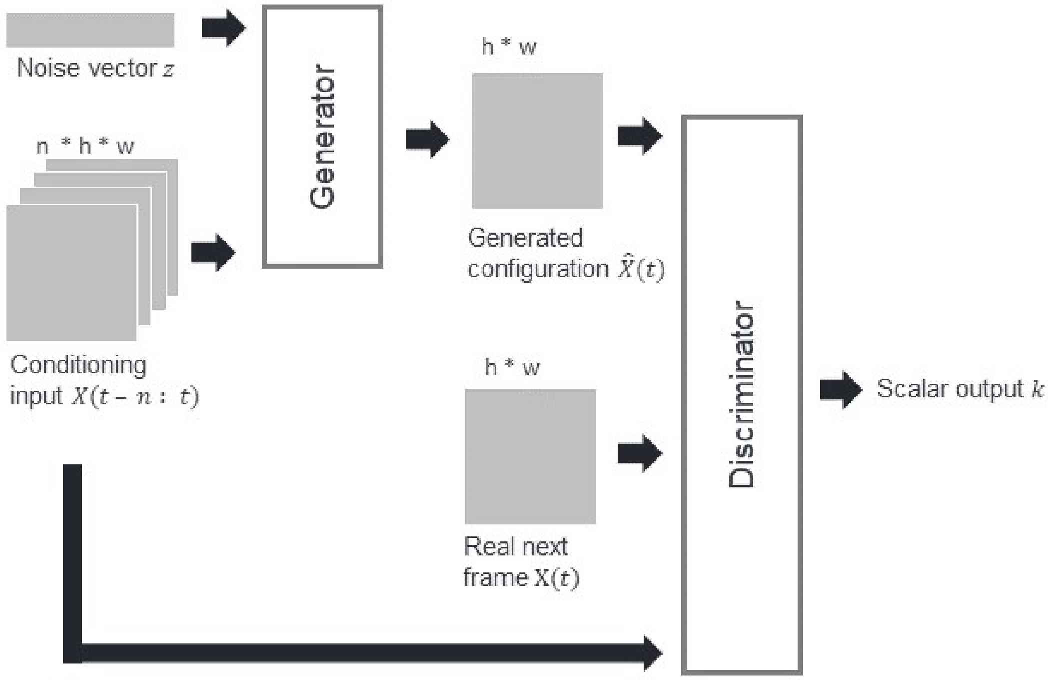

Figure 1 GAN architecture.

5 Results

We implemented the GAN in Python, using the tensorflow library, and tested it on the data of GoL I and II for 50 epochs each with a batch size of 15, using the ADAM optimizer [9] for training both the generator and the discriminator. To track the learning progress, we additionally logged the cross entropy loss of real and generated frames, which is shown in figure 2 for both experiments. With regards to GoL I, the results for GoL I show an almost constant decrease towards 0 for both training and test set, and therefore clearly illustrate a successful learning progress (the high quality of the generated samples is illustrated in figure 3). Thus, the GAN has apparently internalized the underlying rules of the traditional game definition, and was able to generate correct predictions.

However, the results are different for the adapted GoL II. Here, apparently the convolutional layers were unable to successfully encode the adapted, non-proximity based neighborhood definition as defined for a subset of cells. This negative example demonstrates the need for developing and testing alternative network architectures on standardized training data sets to understand the relationships between properties of geographical processes and appropriate network structures. Thus, for instance, in case of GoL II (or other processes with non-proximity based neighborhoods), a deeper network of multiple stacked convolutional layers and larger filter sizes or an attention layer might lead to better results.

6 Conclusion

In this study, we have demonstrated the potential of GAN for understanding the underlying rules of a geographical process directly from its generated data. GANs do not rely on any expert knowledge or theoretical model of the study domain, can be trained end-to-end, and have the ability to generate indistinguishable samples from distributions of any complexity.

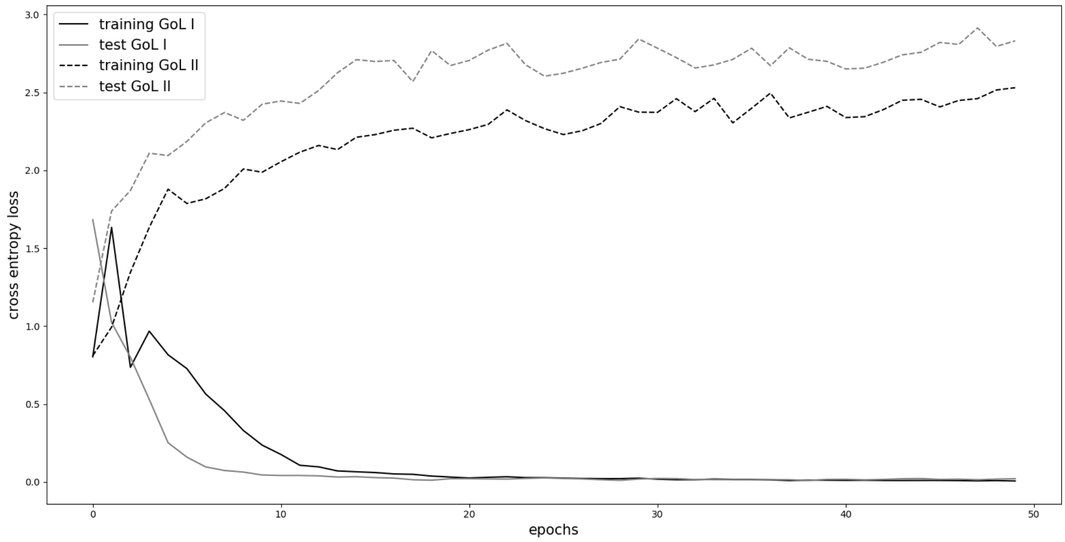

Figure 2 Cross entropy loss for GoL I and II.

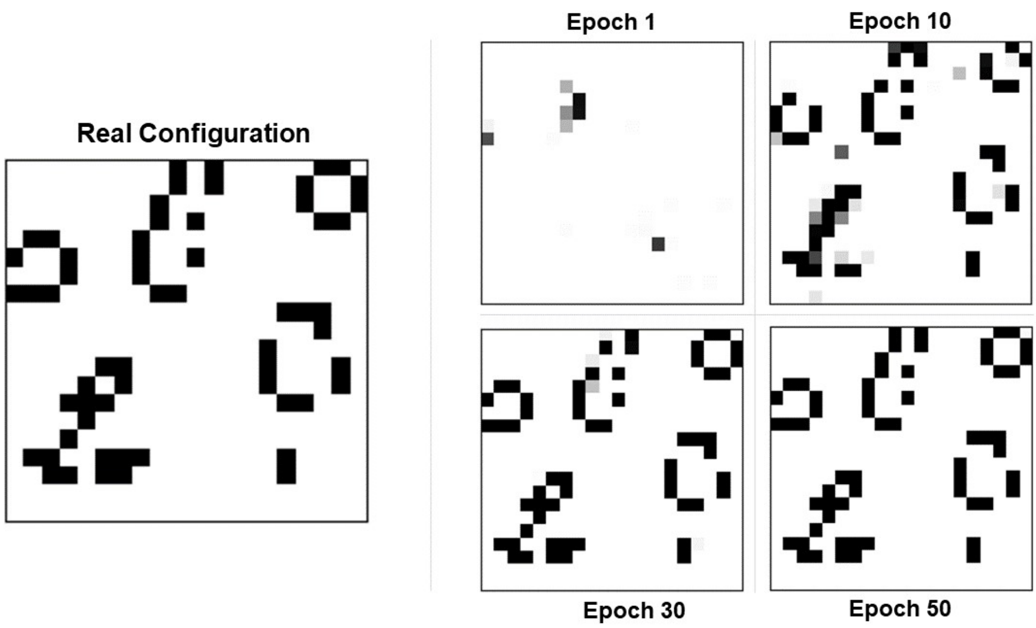

Figure 3 GoL I: Example of the progress of generated samples (right) approximating the real sample (left) until becoming indistinguishable.

Therefore, in our view they are highly promising candidates for simulating geographical processes in general, exploring different scenarios (by conditioning them with different inputs) as well as serve for predictive tasks.

In this preliminary study, we used the GoL as useful abstraction for geographical processes, still, it is clear that its restriction to (few and simple) purely local rules in terms of spatio-temporal interactions represent great simplification compared to real-world processes which are guided by much more complex rules and interactions. Thus, one could expect GAN architectures which were successful on the GoL to fail when presented with real-world spatio-temporal data. Still, however, by manipulating its rules (as we have demonstrated), one could define gradually more complex versions of the game while still maintaining comparable, standard data sets for benchmarking generative models of differing complexities. In general, GANs have no restriction with regards to the complexity of the modeled distribution, i.e., theoretically they can be applied to model any kind of geographical process. Still, more research is needed to evaluate their practical value as an alternative to traditional explanatory or predictive modeling approaches. Additionally, it can be expected that GANs with different architectures will be more or less appropriate to capturing the rules of processes with varying properties. To assess and drive the success of such architectures for general geo-spatial processes will require a set of well understood, plentiful benchmark processes created and utilized by the community.

A downside of GAN is their black box character. Thus, although the network itself has understood and internalized the internal workings of the process, it is challenging to translate that into human-understandable rule descriptions. Still, however, concepts such as transfer learning, where a learned model is transferred and applied to a different task, or attention mechanisms [24] where part of the network's internal reasoning can be made visible (identifying important features for the task), can help to either make explanatory models to a degree obsolete or visualize insights into the derived rules.

References

1. Adrian Albert, Emanuele Strano, Jasleen Kaur, and Marta González. Modeling urbanization patterns with generative adversarial networks. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 2095–2098. IEEE, 2018.
2. Marc P Armstrong. Temporality in spatial databases. In *GIS/LIS 88 Proceedings: Accessing the World*, pages 880–889, 1988.
3. Itzhak Benenson, Paul M Torrens, and Paul Torrens. *Geosimulation: Automata-based modeling of urban phenomena.* John Wiley & Sons, 2004.
4. Claudio EC Campelo, Brandon Bennett, and Vania Dimitrova. Identifying geographical processes from time-stamped data. In *International Conference on GeoSpatial Sematics*, pages 70–87. Springer, 2011.
5. Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.
6. Christophe Claramunt and Marius Theriault. Toward semantics for modelling spatio-temporal processes within GIS. *Advances in GIs Research I*, pages 27–43, 1996.
7. Martin Gardener. MATHEMATICAL GAMES: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:120–123, 1970.
8. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

9 Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

10 Arief Koesdwiady and Fakhri Karray. Improving Multi-Step Traffic Flow Prediction, 2018. `arXiv:1803.01365`.

11 Patrice Langlois. *Simulation of complex systems in GIS*. John Wiley & Sons, 2013.

12 Yann LeCun and Corinna Cortes. MNIST handwritten digit database, 2010. URL: `http://yann.lecun.com/exdb/mnist/`.

13 Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

14 Linchao Li, Shanglu He, Fan Yang, Jian Zhang, and Bin. Ran. Space-mean-speed Prediction Based on the Fusion of Multi-source Data. In *TRB 96th Annual Meeting Compendium of Papers*, 2017.

15 Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion GAN for future-flow embedded video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1744–1752, 2017.

16 Yunyi Liang, Zhiyong Cui, Yu Tian, Huimiao Chen, and Yinhai Wang. A Deep Generative Adversarial Architecture for Network-Wide Spatial-Temporal Traffic-State Estimation. *Transportation Research Record*, page 0361198118798737, 2018.

17 Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

18 Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

19 Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

20 Galit Shmueli et al. To explain or to predict? *Statistical science*, 25(3):289–310, 2010.

21 Willington Siabato, Christophe Claramunt, Sergio Ilarri, and Miguel Ángel Manso-Callejo. A Survey of Modelling Trends in Temporal GIS. *ACM Computing Surveys (CSUR)*, 51(2):30, 2018.

22 Michael Simkin. The hunting of the New Herschel Conduits, 2015. URL: `http://conwaylife.com/forums/viewtopic.php?f=2&t=1599&start=200#p19125`.

23 Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

24 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

25 Andrew J. Wade. Gemini, 2010. URL: `http://conwaylife.com/w/index.php?title=Gemini`.

26 SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

27 Yang Zhang, Xiang Li, and Qianyu Zhang. Road Topology Refinement via a Multi-Conditional Generative Adversarial Network. *Sensors*, 19(5):1162, 2019.