Photonic elementary cellular automata for simulation of complex phenomena

Gordon H. Y. Li¹, Christian R. Leefmans¹, James Williams² and Alireza Marandi¹,²✉

Abstract
Cellular automata are a class of computational models based on simple rules and algorithms that can simulate a wide range of complex phenomena. However, when using conventional computers, these ‘simple’ rules are only encapsulated at the level of software. This can be taken one step further by simplifying the underlying physical hardware. Here, we propose and implement a simple photonic hardware platform for simulating complex phenomena based on cellular automata. Using this special-purpose computer, we experimentally demonstrate complex phenomena, including fractals, chaos, and solitons, which are typically associated with much more complex physical systems. The flexibility and programmability of our photonic computer present new opportunities to simulate and harness complexity for efficient, robust, and decentralized information processing using light.

Introduction
Modern digital electronic computers, which are based on the von Neumann architecture, exhibit extreme hardware complexity in their construction and are composed of billions of transistors engineered in a hierarchical and highly structured manner. Unlike the von Neumann architecture, nature is abound with emergent phenomena and complex systems containing many interacting components following simple rules with no hierarchical control. For example, social insects like ants with only limited local information can collectively self-organize to form global structures¹. This suggests that an alternative and potentially more efficient way to simulate such phenomena is to harness simple and decentralized physical hardware that directly emulates the underlying rules of a complex system.

One class of computational models that can benefit from simple and decentralized physical hardware is cellular automata (CA), which exhibits complex behavior emerging from the local interactions of cells arranged on a regular lattice². CA were introduced in the 1940s to study how self-replication and evolution can emerge in artificial life³ and was later popularized in Conway’s Game of Life⁴, which exhibits self-organizing patterns reminiscent of biological systems. Subsequent landmark studies revealed that CA are also capable of replicating other complex behavior such as fractals⁵, chaos⁶, self-organized criticality⁷, synchronization⁸, and universal computation⁹. Consequently, CA have found utility in modeling a wide range of natural phenomena in physics¹⁰,¹¹, chemistry¹²–¹⁴, and biology¹⁵. Furthermore, CA have important applications in real-world computational problems such as cryptography¹⁶, data compression¹⁷, error-correction¹⁸, simulating traffic flow¹⁹, and developing more robust artificial intelligence²⁰. Owing to their simple formulations, certain CA of interest are computationally irreducible²¹, i.e., there are no analytical shortcuts to evaluate their state after an arbitrary time without resorting to executing the sequential simulation in its entirety. On the other hand, most CA are only implemented as high-level software on conventional computers, resulting in unnecessary overhead. Therefore, it is desirable to seek out physical hardware that better encapsulates the computational principles of CA to enable more efficient simulation. Notable previous attempts to implement physical
systems tailored to perform CA include self-assembling DNA molecules, arrays of nanomagnets, memristor networks, and living slime molds.

In this work, we propose and experimentally implement a photonic computational platform capable of simulating complex phenomena using CA. Compared to other approaches, our photonic platform offers several distinct advantages: (1) the inherently high bandwidth endowed by computing using light offers potentially orders-of-magnitude speed-up in clock rate over the simulation of CA on conventional von Neumann computers, (2) rapid reconfigurability for easy programming of a variety of CA rules enables many different complex phenomena to be observed in the same physical system, and (3) the kind of sparse, local, and shift-invariant connections required for CA are well-suited for this platform. We will demonstrate how even simple photonic hardware can host a wide range of complex emergent phenomena and is capable of sophisticated (or even universal) computation. By exploiting this complexity, we reveal a path toward the next generation of more efficient or robust photonic hardware accelerators for reservoir computing and deep learning.

Results

We focus on the simplest types of CA called elementary cellular automata (ECA). These are discrete-time dynamical systems defined on a 1D lattice of cells with binary states that evolve according to Eq. 1:

$$x_i(t + 1) = f(x_{i-1}(t), x_i(t), x_{i+1}(t))$$

where \(x_i(t) \in \mathbb{Z}_2\) is the state (i.e., dead or live) of the \(i\)th cell at time step \(t\), and \(f: (\mathbb{Z}_2)^3 \rightarrow \mathbb{Z}_2\) is the update rule. Crucially, the rules specifying interactions among cells are computed using only local nearest-neighbor information without reference to the global pattern. Remarkably, not only can the underlying rules be simple, but the initial conditions can also be simple—consisting, for example, of just a single live cell—and yet the collective behavior produced can still be highly complex. The 256 possible ECA rules encapsulate a wide range of complex phenomena and are representative of the four universality classes of increasing complexity introduced by Wolfram.

Here, we experimentally implement ECA in a time-multiplexed photonic system, as shown in Fig. 1a. Cell states are represented using pulses of light produced by a mode-locked laser (MLL) with a fixed repetition rate \(T_R\). The presence of a pulse indicates a live cell, while the absence indicates a dead cell. In this way, the complete pulse train from the MLL represents a synthetic 1D lattice, where each site in the lattice is encoded as a time bin of width \(T_R\) in the pulse train. The pulse amplitude/phase representing the initial cell state is encoded using an electro-optic modulator (EOM), and the pulse is then split between three optical delay lines. These delay lines, which are labeled \(0T_R\) and \(\pm 1T_R\) in Fig. 1a, are chosen such that the \(+(-)1T_R\) delay line is one pulse repetition period longer (shorter) than the \(0T_R\) delay line. With this choice of lengths, light in the \(0T_R\) delay line coherently interferes with its nearest-neighbor time slots in the temporal synthetic lattice. The coherent interference is followed by optoelectronic thresholding to enforce the binary constraint for the updated cell state in the ECA. Finally, the optoelectronic signal is stored on a field-programmable gate array (FPGA), which feeds back the measured cell states by driving the input EOM for the next iteration. By repeating this process for many cycles, we observe the emergence of complex phenomena in the cell states of the ECA. The desired ECA update rule, such as ECA Rule 90 (following the Wolfram naming convention) shown in Fig. 1b, is programmed by tuning the thresholding value and variable optical attenuator (VOA) in each delay line, which represents constant amplitude/phase weights. As shown in Fig. 1c, this rule encoding can be interpreted as an optical dot product followed by a nonlinear thresholding function, which is akin to a single perceptron in the context of artificial neural networks. Therefore, the dynamics of the abstract ECA rule are exactly mapped to the physical time evolution of the photonic simulator. We note that similar time-multiplexing techniques with delay dynamics have been used for photonic reservoir computing, coherent Ising machines, simulating topological lattices, and optoelectronic oscillators with time-delayed feedback. Analogous results can be obtained using other multiplexing methods, such as frequency or spatial multiplexing.

First, one of the most striking patterns that emerge in CA are fractals, which are often self-similar geometric shapes that appear the same at any scale. Fractals are ubiquitous in nature and occur in a diverse range of physical phenomena, including the rings of Saturn, snowflakes, and fault geology. ECA Rule 90, defined in Fig. 1b, provides a simple model for fractal formation and self-replication. The local update rule can be expressed succinctly in terms of Boolean algebra as \(x_i(t + 1) = x_{i-1}(t) \oplus x_{i+1}(t)\), where \(\oplus\) denotes the exclusive-or (XOR) logical operation. Thus, for this specific rule, the iterated cell state depends only on the states of its two neighbors.

The fractal pattern is an emergent property of the nonlinear dynamics in the photonic computer, rather than being imposed on the system by an external ordering influence such as explicit geometric constructions in previous studies of photonic fractals. We show the experimentally measured space-time equivalent diagram of ECA Rule 90, starting from a single live cell, in Fig. 2. The position of a cell in space (left-to-right) is represented by the pulse number in the synthetic temporal dimension, and the discrete-time step (top-to-bottom) is defined...
according to Eq. 1. The color of each cell is determined by the normalized peak pulse intensity before thresholding. We see that the space-time diagram is in the shape of the well-known Sierpinski Triangle. This fractal can be constructed by recursively subdividing an equilateral triangle into four smaller equilateral triangles and removing the central triangle. It is characterized by a non-integer Hausdorff or fractal dimension of $\log_3/\log_2 \approx 1.585$. The self-similarity of the fractal shape persists down to the cellular scale, as shown in Fig. 2b, and can be seen in the time traces shown in Fig. 2c, where individual light pulses represent each live cell. In this case, the middle optical delay line ($0T_R$ in Fig. 1a) can be ignored since the iterated cell state does not depend on its current state. Experimentally, this allows us to achieve an excellent extinction ratio between pulse peaks for live and dead cells and indicates that the ECA Rule 90 is implemented as intended.

Next, we investigate ECA Rule 30, defined in Fig. 3a, which is categorized as a member of class 3 CA according
to Wolfram’s universal complexity classes. These are CA that produce chaotic and seemingly random patterns, although some small-scale structures are present\(^2\). Remarkably, ECA Rule 30 is one of the simplest known systems to exhibit chaos\(^6\). We experimentally demonstrate such a chaotic behavior of ECA Rule 30 on the same simple photonic hardware in Fig. 3b starting from a random initial condition. In this case, there is greater variability in the peak pulse intensities compared to Rule 90 due to the lower interference visibility between three optical delay lines. However, the optoelectronic thresholding is still adequate to ensure the intended operation of ECA Rule 30. A necessary (but not sufficient) condition for chaos is sensitivity to initial conditions. Figure 3c shows the space-time diagram starting from the same initial condition as Fig. 3b but with one cell inverted. The region of differences between the two patterns grows linearly to the right with Lyapunov exponent \(\lambda_R = 1\) and asymptotically linearly to the left with Lyapunov exponent \(\lambda_L \approx 0.24\), hence implying an exponential divergence in the cell configurations over time and sensitivity to initial conditions. Other necessary conditions for chaos, such as non-periodicity and topological mixing, have also been verified empirically\(^6\).

Due to the simplicity of ECA Rule 30, it can be used as an efficient pseudo-random number generator. This can be accomplished, for example, by taking the sequence defined by the states of the central cell as it evolves in time, i.e., the middle column of the space-time diagram. Therefore, the initial condition acts as the seed.

\(^{2}\) Li et al. Light: Science & Applications (2023) 12:132
Importantly, ECA Rule 30 is highly nonlinear and computationally irreducible, unlike ECA Rule 90, which is linear (modulo 2) and amenable to algebraic analysis. Indeed, detailed statistical analysis of the sequences produced by ECA Rule 30 shows that it is both a fast and safe random number generator. Unlike previous photonic random number generators relying on quantum processes or other continuous fluctuations, our system is pseudo-random, which means it is deterministic and repeatable given the initial seed. This is often useful in practice to reliably reproduce results in applications such as Monte Carlo simulations, stream ciphers, and generative adversarial networks. We note that ECA Rule 30 was previously demonstrated using free-space optics; however, this implementation encoded cells on pixels of 2D liquid-crystal screens, which introduced some redundancy. In contrast, our approach more faithfully implements the 1D lattice for ECA, can be easily extended to an arbitrary number of cells, and is easily programmable to implement more than just a single rule.

Finally, we study class 4 CA, which involves a mixture of order and randomness with localized structures that move and interact in complicated ways. A well-studied example of this is ECA Rule 54, defined in Fig. 4a, which can be interpreted as a discrete analog of excitations in an active nonlinear medium with mutual inhibition.

In this case, the mobile self-localizations called gliders appear on a stable periodic background called the ether. Gliders behave like solitons in many regards. However, while optical solitons usually arise due to a balance between nonlinear and linear dispersive effects, we have demonstrated optical soliton-like behavior in a synthetic temporal lattice with only simple binary rules. Despite its simplicity, our system captures physically relevant features since a reversible extension of ECA Rule 54 has produced insightful results in non-equilibrium statistical mechanics and generalized hydrodynamics. By properly programming ECA Rule 54 in our photonic simulator, we experimentally demonstrated a glider collision, shown in Fig. 4b, whereby gliders emerge after the collision with the same shape and velocity but with a phase shift, which is characteristic of soliton collisions. Such glider collisions can be used to construct logic gates and Universal Turing Machines for unconventional computing. Furthermore, we also observed a glider gun, shown in Fig. 4c, in which a higher-order localization produces lower-order gliders akin to the process of soliton fission. Conversely, a glider black hole, shown in Fig. 4d, looks like the process of soliton fusion. Therefore, we have demonstrated a diverse range of glider and soliton interactions in our simple photonic computational platform, which can help unlock new methods of optical information processing.

**Discussion**

In our current experiments, we performed the weighted linear summations in the optical domain and applied nonlinearity digitally for convenience in demonstrating the photonic ECA concept. However, this incurs additional overhead due to the optoelectronic and digital-to-analog conversions performed. This overhead can be reduced by using fully analog electrical circuits, but the photodetector bandwidth still ultimately bottlenecks the clock speed of CA. To overcome these electronic limitations, we propose an integrated all-optical implementation of photonic CA so
that the self-evolution of the CA occurs entirely in the optical domain. An example on-chip implementation of photonic CA is shown in Fig. 5. This implementation is based on the same time-multiplexed architecture as our current experiment, but it utilizes a monolithic thin-film lithium niobate platform\textsuperscript{59} to increase computation speed and energy efficiency by potential orders of magnitude. As a measure of these speed and energy enhancements, we consider the energy-time product, which captures both the energy and speed of the device operation. All-optical rectification and switching operations\textsuperscript{60,61} can be performed using nonlinear waveguides in thin-film lithium niobate with energy-time products on the order of $10^{-27}$ J·s, which is more than three orders of magnitude better than the kind of optoelectronic thresholding performed in the current experiments. This can be followed by an optical feedback loop to enable terahertz clock rates (Supplementary Information Section 4) that are unattainable by digital electronics. Moreover, the VOAs of our current implementation can be replaced by static integrated EOMs to provide greater control and reproducibility in setting amplitude weights for specific rules. Similarly, other photonic components in our system can be replaced by their simpler and higher-performance integrated counterparts.

In summary, we have demonstrated a special-purpose photonic computational platform utilizing a synthetic temporal dimension and simple hardware components capable of simulating a wide range of complex phenomena. Simple rules based on local shift-invariant interactions are used to effectively implement different ECA. Our decentralized and non-von Neumann photonic computer can be programmed to represent different rules and initial conditions for the light pulses due to the flexibility and rapid reconfigurability afforded by our hardware system. A range of important complex phenomena, including fractals, chaos, and solitons, are shown on the same hardware. We focused only on three ECA rules (90, 30, and 54) that exhibit some representative complex
phenomena. However, a slight modification to the present photonic hardware mapping can allow for the implementation of all 256 possible ECA rules (see Supplementary Table IV). Future work can involve generalizing the time-multiplexed photonic network to exploit spatial or frequency multiplexing techniques for greater synchronous parallel cell updates or implementing other types of CA, including filter CA, reversible CA, and stochastic CA. This can enable the study of experimentally challenging complex dynamics in kinetic critical phenomena, Ising models, and lattice Boltzmann models. Furthermore, achieving complexity from simple photonic hardware is an important first step toward harnessing this complexity for efficient and robust artificial intelligence, for example, in reservoir computing and deep learning. Our results can inspire a path for special-purpose photonic computers enabling ultrafast low-power operation for critical real-time and edge-computing applications and new information processing strategies using light.

### Materials and methods

#### Experimental setup

For a more detailed picture of our experimental setup, please see Supplementary Information Fig. 1. A modelocked laser (MLL) is used that outputs femtosecond optical pulses with a center wavelength of 1550 nm and a repetition period of $T_R = 4$ ns. Then, the pulses are stretched to $\sim 5$ ps with a 200-GHz Channel 34 filter to reduce the effects of dispersion. After the pulses are stretched, 10% of the power is tapped with a 90:10 optical fiber splitter and sent directly to a 600-MHz bandwidth photodetector. The RF output of the detector passes through a 300-MHz low pass filter, which isolates the 250 MHz component of the signal. This signal acts as a clock for the FPGA (Zynq UltraScale+ RFSoC), which generates the modulator driving signals for the EOM in the experiment. Deriving the FPGA’s clock directly from the optical pulse train eliminates any timing drift between the optical path and electronic signals. The 90% of the optical power that is not used to clock the FPGA is instead sent through two consecutive intensity modulators (IMs). The first IM converts the uniform input pulse train to a binary string that contains either an initial condition or the previous state of the ECA under study. The second IM helps to achieve a better extinction ratio for the zeros in these binary strings.

[Fig. 5 Lithium niobate nanophotonic cellular automata. The simplicity of the photonic hardware components for simulating complexity can be maximized by on-chip integration with lithium niobate nanophotonic circuits. For example, integrated EOMs offer greater performance, and a periodically-poled nonlinear waveguide can enable efficient all-optical thresholding and feedback. This simple nanophotonic circuit can yield orders of magnitude improvement in the speed and energy efficiency for simulating complexity in CA]
The pulses are split at a final 50:50 splitter, and the signal is then measured on a fast 5-GHz photodetector and a slow kHz photodetector. The RF output of the slow detector is sent to the stabilization electronics for the delay lines, while the RF output of the fast detector is recorded on an oscilloscope. The optoelectronic signal is thresholded electronically to produce binary states, which are then sent to and stored on the FPGA, which uses a digital-to-analog converter (DAC) to convert the array into an RF pulse pattern for the next ECA iteration.

**Experimental procedure**

Setting the desired ECA rule involves adjusting the relative intensities and phases between the three delay lines. VOAs are used to adjust the intensities in the lines by detuning the coupling in the free space delays shown in Supplementary Information Fig. 1, and the relative phases are set to either 0 or $\pi$ by changing the feedback signals from the PIDs used to stabilize the $\pm 17\tau$ delay lines. A relative phase of 0 represents constructive interference between two delay lines, and conversely, a relative phase of $\pi$ represents destructive interference. Therefore, the result of the $\pm 17\tau$ delay lines, tuning the VOAs, and setting relative phases can be summarized as:

$$y_i(t) = a_{-1}x_{i-1}(t) + a_0x_i(t) + a_1x_{i+1}(t)$$

where $x_i(t) \in \mathbb{Z}_2$ is the amplitude of the $i^\text{th}$ light pulse in the $t^\text{th}$ iteration before being split into the delay lines, $y_i(t)$ is the amplitude of the light pulse after recombining delay lines, and $\{a_{-1}, a_0, a_1\} \in [-1, 1]$ are the losses set by the VOAs and phases representing constant linear weights. The light pulse amplitude $|y_i(t)|$ is converted to an intensity $|y_i(t)|^2$ after passing through the photodetector, and then optoelectronic thresholding performs the function:

$$x_i(t+1) = H\left(|y_i(t)|^2 - b\right)$$

where $H(x)$ is the Heaviside step function, $b \in \mathbb{R}$ is the thresholding value, and $x_i(t+1)$ is the output result to be reinserted as the light pulse amplitude for the next iteration. Therefore, any light intensity $|y_i(t)|^2 < b$ represents a dead cell, and conversely, any light intensity $|y_i(t)|^2 > b$ represents a live cell. The particular mappings for each ECA rule studied in “Results” is given in Supplementary Information Section 3.

In the present experiments, we considered effectively infinite lattices by padding both sides of the initial conditions with large numbers of zeros that exceeded the number of measured iterations for each experiment. It is also possible to implement other types of boundary conditions, such as closed boundaries, by imposing constant boundary cell states that do not update or periodic boundary conditions by turning the 1D lattice into a ring geometry.

**Acknowledgements**

The authors gratefully acknowledge support from ARO grant no. W911NF-23-1-0048, NSF grant no. 1846273 and 1918549, AFOSR award FA9550-20-1-0040, and NASA/JPL. The authors wish to thank NTT Research for their financial and technical support.

**Author contributions**

All authors contributed to this manuscript.

**Data availability**

The data used to generate the plots and results in this paper are available from the corresponding author upon reasonable request.

**Code availability**

The code used to analyze the data and generate the plots for this paper is available from the corresponding author upon reasonable request.

**Conflict of interest**

G.H.Y.L. and A.M. are inventors on a US provisional patent application filed by the California Institute of Technology (application number 63/441,387) related to the work presented here. The remaining authors declare no competing interests.

**Supplementary information**

The online version contains supplementary material available at https://doi.org/10.1038/s41377-023-01180-9.

Received: 8 December 2022 Revised: 28 April 2023 Accepted: 8 May 2023
Published online: 30 May 2023

**References**

1. Bonabeau, E. et al. Self-organization in social insects. Trends Ecol. Evol. 12, 188–193 (1997).
2. Wolfram, S. Cellular automata as models of complexity. Nature 311, 419–424 (1984).
3. von Neumann, J. & Burks, A. W. Theory of Self-Reproducing Automata (University of Illinois Press, 1966).
4. Gardner, M. The fantastic combinations of John Conway’s new solitaire game Life. Sci. Am. 223, 120–123 (1970).
5. Wolfram, S. Statistical mechanics of cellular automata. Rev. Mod. Phys. 55, 601–644 (1983).
6. Wolfram, S. Random sequence generation by cellular automata. Adv. Appl. Math. 7, 123–169 (1986).
7. Bak, P., Tang, C. & Wiesenfeld, K. Self-organized criticality: an explanation of the 1/f noise. Phys. Rev. Lett. 59, 381–384 (1987).
8. Balzer, R. An 8-state minimal time solution to the firing squad synchronization problem. Inf. Control 10, 22–42 (1967).
9. Cock, M. Universality in elementary cellular automata. Complex Syst. 15, 1–40 (2004).
10. Wolf-Gladrow, D. A. Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction (Springer, 2004).
11. Chopard, B. & Droz, M. Cellular automata model for the diffusion equation. J. Theor. Biol. 123, 188–193 (1986).
12. Gerhardt, M. & Schuster, H. A cellular automaton describing the formation of dead cell states. J. Theor. Biol. 109, 123–133 (1984).
13. Ermentrout, G. B. & Edelstein-Keshet, L. Cellular automata approaches to biological modeling. J. Theor. Biol. 160, 97–133 (1993).
14. Chowdhury, D. R. et al. Design of CAECC-cellular automata based error correcting code. IEEE Trans. Comput. 43, 759–764 (1994).
