Optimising Weakly Connected Chemical Oscillators in a Hybrid Digital–Chemical Platform via Generative AI Surrogate Modelling

Juan Manuel Parrilla Gutierrez
Glasgow Caledonian University
School of Science and Engineering
The University of Glasgow
School of Chemistry
juanma.parrilla@gcu.ac.uk

Abhishek Sharma
The University of Glasgow
School of Chemistry
Abhishek.Sharma@glasgow.ac.uk

Soichiro Tsuda
The University of Glasgow
School of Chemistry
soichiro.tsuda@gmail.com

Leroy Cronin*
The University of Glasgow
School of Chemistry
lee.cronin@glasgow.ac.uk

Abstract Herein, we present a 3D-printed platform consisting of a rectangular 2D-array of interconnected cells containing the Belousov–Zhabotinsky (BZ) reaction. This reaction can be made to oscillate between two states to simulate the binary codification of digital electronics. Within the platform each cell contains a magnetic stirrer that can be individually stirred to control the local oscillations of the BZ reaction in that cell. Moreover, all the cells are also weakly interconnected through the common medium, and their collective oscillatory dynamics can be used to perform heterotic computations. The 3D-printed vessel can be fabricated using different architectures to, for example, define how the cells are connected, and thus controlling how the oscillations propagate between them. We took advantage of these features to simulate the "AND", "OR", and "XOR" logic gates. To increase the experimental throughput of our platform and further investigate its computability capabilities, we modelled it using Generative Artificial Intelligence (AI). Then, the AI model was used as an *in-silico* surrogate, in combination with a Genetic Algorithm, to explore and optimise the behaviour of our platform towards user-defined objectives. The results presented here provide a proof of concept for combining hybrid digital–chemical platforms with Generative AI surrogate modelling to explore unconventional computation, which might potentially enhance future Artificial Life implementations more effectively than current silicon-based advancements.

Keywords
Unconventional computation, chemical computer, chemical oscillator, generative AI, Hybrid Digital–Chemical, 3D-printing

1 Introduction

Nature has developed systems with enviable capabilities to process complex information from environments that remain challenging to replicate in conventional digital computers. For example, a bacterium can navigate an environment full of thermal noise, despite its small size (Sourjik, 2004); or the human brain can process information by integrating sensory inputs and converting them into a spatial firing pattern (Deco & Kringelbach, 2017). Even non-living chemical systems can be implemented to perform computations (Baltussen et al., 2024; Qian & Winfree, 2011).

The idea of harnessing such nonlinear natural dynamics for computing can be dated back to the 1980s (Conrad, 1985) and has been investigated in various forms, such as quantum states (Zhu & Wang, 2003), enzymes (Baron et al., 2006), DNA (Prokup et al., 2012; Yang et al., 2024), bacterial

* Corresponding author.

© 2026 Massachusetts Institute of Technology

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

population (Gardner et al., 2000) and chemical reactions (Adamatzky et al., 2005; Fletcher et al., 2025). In particular, the Belousov-Zhabotinsky (BZ) chemical reaction has been widely studied as media for unconventional computing systems through exploiting the reaction-diffusion nature of its oscillations (Gorecki et al., 2015; Guzowski et al., 2016; Toth et al., 2008). The BZ reaction medium, either in continuous bulk solution or compartmentalised in droplets, has been used to perform information processing tasks, such as image processing (Rambidi et al., 1994), optimisation algorithms (Guo et al., 2021; Krasecki et al., 2023), path planning (Adamatzky & de Lacy Costello, 2002), robot control (Tsompanas et al., 2019), logic gates (Tóth & Showalter, 1995), the emulation of complex circuits (Sun & Zhao, 2013), or the implementation of Turing Machines (Dueñas-Díez & Pérez-Mercader, 2019).

Moreover, the BZ reaction has been extensively used as a medium for the implementation of Boolean logic. The collisions of wave fragments in BZ medium, for example, has been experimentally verified to reproduce boolean logic such as the AND, OR and NOT gates, among other information transfer processes. These results were obtained by investigating how the angle of the collisions impacted the evolution of the wave fragments (Costello & Adamatzky, 2005). This same approach has also been extended to fully implement AND, OR, and NOT gates, as well as more complex logic circuits, using geometrically constrained interactions, or by inducing oscillations using light sensitive BZ medium (Stevens et al., 2012; Toth et al., 2009). Using the same approach focused on the collision of wave fragments, the authors also implemented Fredkin and Toffoli gates (Adamatzky, 2017). In further extensions of this research they also managed to implement higher-level digital components such as binary full adders and decoders (Adamatzky, 2015; Sun & Zhao, 2013). Alternative architectures, including encapsulated discs, droplets, and vesicle-based systems, have further demonstrated logical universality and reconfigurability using the BZ reaction (Holley et al., 2011).

However, exploiting the continuous (analog) nature of chemical and biological dynamics for information processing presents challenges in observability and control. In many purely chemical implementations, system states are inferred indirectly, and control is usually limited to initial conditions or static geometries, making it difficult to reprogram the system in real-time or to reliably extract outputs. In this work, we therefore focus on a hybrid digital–chemical approach, where a digital interface is used to enable real-time observation and control of the underlying chemical dynamics (McEvoy & Correll, 2015). This allows continuous mapping between analog chemical states and discrete representations, as well as real-time actuation. This could potentially enable the system to be controlled in a programmable and adaptive manner (Kendon et al., 2015).

To further extend the capabilities of this hybrid system, particularly in terms of experimental throughput and exploration of the accessible dynamical space, we focused on computational modelling. For the BZ reaction, the most widely known theoretical model is the Oregonator model (Hsu et al., 1994), although more recent advances have focused on empirically derived models (Karimov et al., 2023). However, these models often fall short in capturing the full dynamics of real world experimental systems, such as the BZ platform we developed in previous research (Cronin et al., 2019). To model platforms such as this one, we directed our attention towards recent advances in the field of Generative AI. Generative AI has gained mainstream recognition by achieving excellent results in tasks such as language (Vaswani et al., 2017) or image generation (Ho et al., 2020). Nevertheless, it has also achieved outstanding results modelling physics (Arnold et al., 2024; Cachay et al., 2024) and chemistry phenomena (Joung et al., 2025; Park et al., 2025).

Herein, we show a 3D-printed platform used to discretise the BZ reaction medium into individual cells (Figure 1). The system is designed to be akin to a 2D cellular automata: Each cell can be individually addressed to trigger the excitation of the BZ reaction by activating a magnetic stirrer (corresponding to the "1" or "ON" of digital electronics). In addition, by changing the size of the gap connecting neighbouring cells, it is possible to control the propagation of BZ excitation waves (Figure 2). If the gap is wide enough, the whole BZ medium generates coherent excitation wave patterns depending on the input states. In this research we demonstrate how these wave patterns can be used to perform computations by simulating logic gates. Moreover, to further investigate its computational capabilities, we developed an *in-silico* model of our platform using Generative AI. This

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

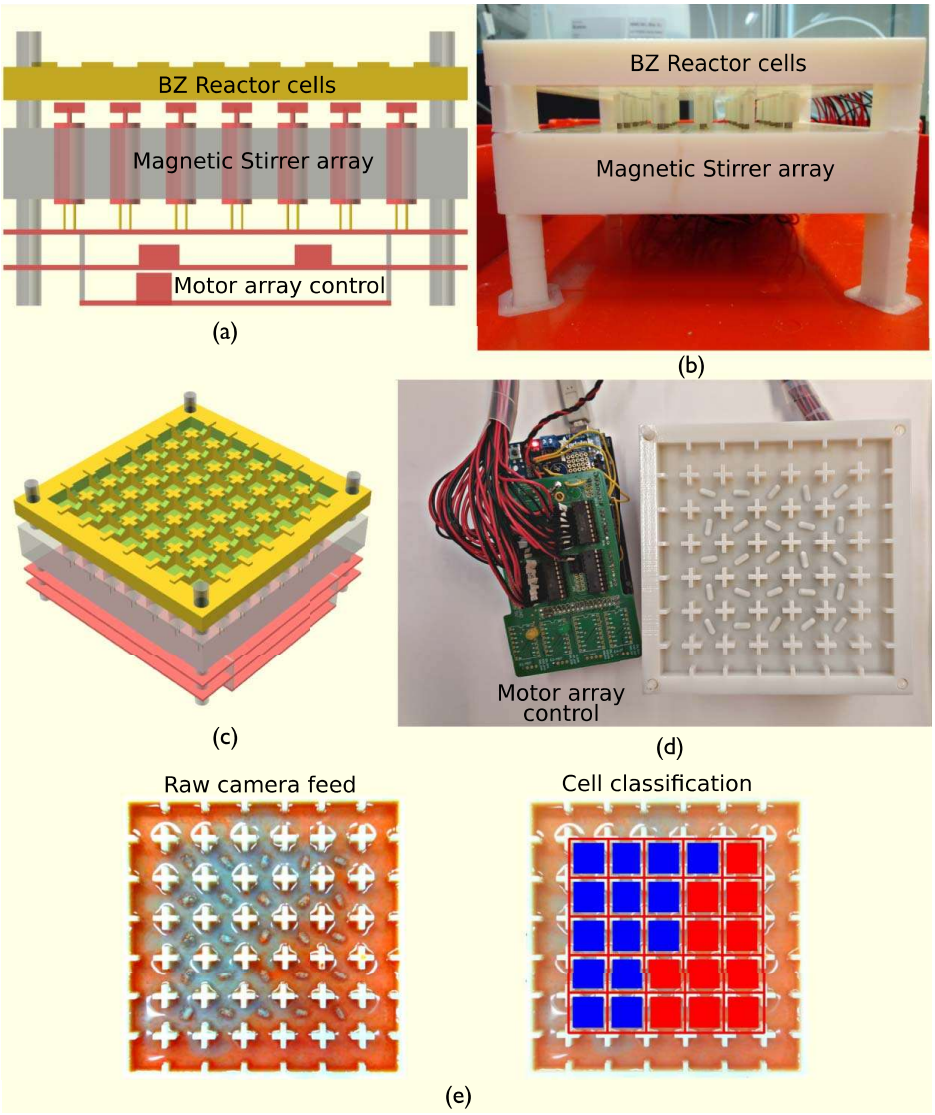


Figure 1. 3D-Printed heterotic computing platform. (a) Outline of the platform, highlighting its main components: the PWM generator, the Arduino board micro-controller, the magnetic stirrer array holding 25 geared DC motors, and the BZ reactor cells where the BZ reaction is placed. (b) Side photograph of the actual platform. Between the stirrer array and the BZ cells, there it can be seen the motor shafts, which have attached a mount with two magnets. When the motors rotate, the stir bars (seen in "D") also rotate. (c) Diagonal view of "A" where the grid cell design can be seen. (d) Photograph of the Arduino board, the PWM generator and the BZ arena. Within the BZ arena the stirrer bars can be seen. When the motors underneath rotate, the stirrer bars also rotate. (e) Pictures of the BZ reactor cells from the camera point of view. Left: Raw image as received from the camera. Right: Processed image where the cells are classified as either red or blue, depending on the actual colour of the BZ reaction in that particular cell.

model was then combined with an optimisation algorithm to explore how the states of the BZ reaction could be controlled via the input stirrers, and to optimise the configuration of these stirrers to manipulate the BZ medium into user-defined behaviours. All optimisation experiments discussed in this work were performed within the surrogate model and were not experimentally validated on the physical platform.

The hybrid nature of the platform plays a central role in its functionality. The digital layer enables real-time classification of the chemical state through computer vision, providing a direct mapping between chemistry and digital binary representations. At the same time, programmable actuation through individually controlled stirrers allows for real-time input to be applied during operation. This combination of observability and controllability is difficult to achieve in purely chemical systems, and

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

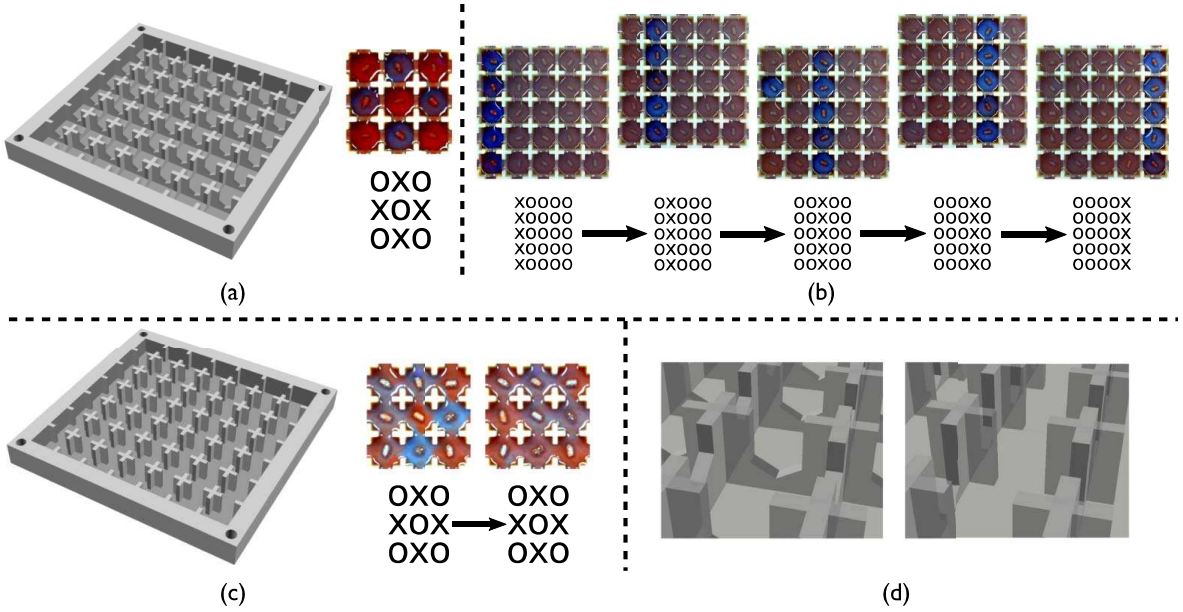


Figure 2. Controlling the oscillations within the BZ medium. In this figure, every video frame is accompanied by a matrix of "x" and "o". This matrix indicates if in that given frame a motor was enabled ("x") or disabled ("o"). (a) Left: Arena with a v-shape opening between cells. With this design, BZ oscillations did not propagate to neighbouring cells. Right: Only the four cells in blue oscillated, and the oscillations did not propagate to other cells. (b) With a v-shape opening between cells, the platform could be used to display images like in a screen. As an example, a sequence of patterns were inputted to the BZ platform, one after the other. Initially only the left most column was enabled, followed by the following ones in successive steps, until arriving to the rightmost column. Only the cells with enabled motors oscillated, and oscillations did not propagate between cells. (c) Left: Arena with a full opening between cells. With this design, BZ oscillations did propagate between neighbouring cells. Right: Initially, the same four cells as before oscillated, but eventually the oscillations propagated to the centre cells - which did not have its motor enabled. (d) Close-up comparing the v-shape opening (from a), with the full opening between cells from (from c).

we considered in this research that this hybrid architecture might be a possible way forward for implementing adaptive computational behaviours.

2 3D-Printed Heterotic Computing Platform

Our 3D-printed heterotic computing system was designed using CAD tools and manufactured with a commercial 3D-Printer. The system consists of three components (Figure 1): (1) The BZ reactor cell grid that was 3D-printed and can be customised to any desired geometry. In our experiments, we used a 5-by-5 grid of cells. Each cell contained a stir bar (8-by-3 mm) to excite the BZ reaction within the cell. (2) A magnetic stirrer array to control and actuate the stir bars in the BZ reactor cells. We used 25 geared DC motors (6V 200RPM) assembled in a 5 by 5 grid. The motors were placed directly underneath the cells. Each motor had a magnet attached to its shaft, so that when the motor rotated, it also actuated the stir bar in the cell above. (3) An Arduino-based electronics controller interfacing a computer and the magnetic stirrer array. The rotation speed of each stir bar could be individually controlled using pulse width modulation (PWM) signals sent from the computer via serial connection. This PWM signal was generated using Arduino Mega, and the shield from Adafruit '16-Channel 12-bit PWM/Servo shield–I2C interface'. A computer connected to the Arduino board through USB defined the different PWM signals through a Python script.

The BZ reaction medium placed in the cell grid was monitored by a camera (LifeCam Cinema, Microsoft) mounted on top of the system (Figure 1e), which operated at 800-by-600 pixels and 30 FPS. The camera was connected to a computer which analysed the BZ reaction in real time, and classified each cell as non-excited state (red) or excited state (blue). This information was then

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

analysed by the computer, and sent back into the platform by individually controlling each of the geared motors via the Arduino board in order to achieve the desired functionality. The classification of each excitation state of the BZ reaction was performed using a "Support Vector Machine" (SVM) which was pre-trained using a manually labelled dataset. Details on the implementation and use of the SVM have been reported in a previous publication (Cronin et al., 2019), which this work builds upon. The SVM was trained on a manually labelled dataset of 2,429 samples, split into a training set with 2,000 samples, and test set with 429 (a 82/18 ratio). The classifier achieved an accuracy of 96% on the test set, with a precision of 96% and recall of 93% for the "blue" (excited) class.

3 Controlling the Oscillations of the BZ Reaction

Ferroin was selected as the sole catalyst due to its ability to induce oscillations, coupled with its clear transition in colour from red (reduced ferroin) to blue (oxidised ferroin) upon oxidation, allowing for optical tracking. To obtain the ferroin indicator, a stock solution of 1 M was prepared by dissolving 5.406 g of 1,10-phenanthroline in 10 mL water and adding 2.60 g ferrous sulfate hexahydrate while stirring. The other chemical components were sulphuric acid (1 M H_2SO_4 was prepared by taking 5.6 mL conc. H_2SO_4 (96%, 18 M) and adding to water to reach a total volume of 100 mL), malonic acid (1 M solution was prepared by dissolving 10.4 g malonic acid in 100 mL of water) and potassium bromate (0.5 M solution was prepared by dissolving 8.35 g $KBrO_3$ in 100 mL of 1 M H_2SO_4).

To control the reaction, we utilised the understanding that large-scale oscillations of the BZ reaction can destabilise and turn chaotic without consistent mixing (Hsu et al., 1994; Karimov et al., 2023). In other words, by activating the stirrer positioned within each individual BZ "cell" in the 3D-printed grid, we can control the stimulation of that specific cell. Therefore, without stirring in the cells, the BZ oscillations were either short-lived or inhibited, and the BZ medium remained in the reduced state, displaying a uniform red colour associated with reduced ferroin. Once it was stirred, sustained oscillations were observed, characterised by periodic transitions between the red (reduced ferroin) and blue (oxidised ferroin) states. Once the stirring stopped, the oscillation amplitude gradually decayed, leading to progressively weaker oxidation events until the system returned to a reduced state characterised by a uniform red colour. This way, in our system the BZ reaction acted as a forced-damped oscillator: When the stirrers were active it acted like a forced oscillator with stable amplitudes, but once the stirring was stopped, it became a damped oscillator and amplitude of the oscillations fell over time.

To stir the cells and consequently induce patterns in the resulting oscillations, the platform described in the previous section (Figure 1) was employed. Each cell housed a stirring bar and was positioned directly over a motor equipped with a set of opposing magnets attached to its shaft. This way, the motors could be turned on and those cells were stirred, causing the BZ reaction mixture in the cell to oscillate. The pattern of stirring in the grid described by the motors could then impart a pattern on the chemical system, where stirred cells were more likely to oscillate and then drive oscillations in neighbouring, unstirred, cells.

Using additive manufacturing (3D-printing) to construct grids of discrete, but fluidically connected, square cells, enabled us to quickly test different designs. The grids were manufactured in white material to allow the colour of each cell to be easily monitored. The basic starting design consisted of a monolithic rectangular block into which a square arena was inset, therefore just one big cell. This was then divided into equal sized cells by printing walls at equal spacing, with cut-outs in each wall to allow fluidic connection. The initial experiment performed was to determine a suitable cell design that would continue to produce bulk oscillations over the whole platform, while being sufficiently large to be imaged consistently. The best results were obtained using a grid with 7-by-7 cells. Each cell was of size (length, width, height) 16-by-16-by-11 mm, where the cut-out accounted for approximately a third of each connecting wall. Of the 7-by-7 grid, only the middle 5-by-5 cells were used for experiments in order to avoid cells on the edges (with fewer neighbours) which might oscillate differently.

In order for computation to occur, information, such as the excitation states of the BZ reaction, had to propagate between cells, akin to the influence neighbouring cells exert in a cellular automaton.

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

In the first grid prototype, the cells had a "v-shape" opening between them (Figure 2a). The BZ reaction volume used was 70 ml, which filled the arena to three quarters of its height, well above the "v" opening. With this design, it was found that the oscillations did not propagate to neighbouring cells and the platform acted similarly to a "display screen", where only the cells that were enabled (their motor was on and thus the stir bar stirred the fluid) oscillated in blue, while the other ones remained red. Therefore, with this design there was a one-to-one mapping between the cells that were stirred and the amplitude of the BZ oscillations.

Using this design, when four cells were enabled in a pattern around an unstirred central cell, the four activated cells flashed in blue, but the oscillation never expanded to the cell in the centre (Figure 2a Left). To showcase the functionality of our platform as a display screen, we used a series of static activation patterns displayed in a sequence (Figure 2b), activating the leftmost column first, followed by the subsequent columns in sequence, with oscillations adhering to the specified pattern.

In order to facilitate cell-to-cell communication, we completely removed the "v-shaped" part of the opening, leaving only the corners of each cell to define the cell itself (Figure 2c). In this manner, as demonstrated in the 4-way neighbourhood test, from an activated BZ cell reached its neighbouring cells when stirred. This way, we could activate a previously inactive cell by stirring its neighbouring cells (Figure 2c Right). Once a cell started to oscillate, its motor would be enabled to keep the oscillations. The camera placed above the platform created a closed-loop and dynamic readout was utilised to activate the cells based on observed oscillations: If BZ cell as read from the camera was classified as "on" (blue) by the SVM, then the stirrer in that cell would be activated, and that cell will continue to oscillate. These results might suggest that the configuration of the opening between cells impacts how the BZ oscillations propagate between them. Following this, in related research we investigated the role of the opening by placing an extra stir bar on it, which would enable or disable the connection between cells (Sharma et al., 2024). Using this different architecture, we implemented 1D and 2D Chemical Cellular Automata, as well as demonstrating hybrid probabilistic logic for solving combinatorial optimisation problems.

4 Visualising the Oscillations

As previously explained, all of our experiments were recorded using a camera placed on top of the platform (see Figure 1e). The camera recorded at a resolution of 800 × 600 pixels and 30 FPS. Given that most experiments lasted 30 minutes, each video contained approximately 54,000 frames.

To better visualise and analyse the results, the videos were flattened into a single image, which we refer to as a "timeline". To construct this timeline image, we first considered that our platform consisted of 25 cells arranged in five rows and five columns. For each frame, we selected five vertical strips with a width of one pixel, each corresponding to one column of the platform. For example, the first 1-pixel strip covered the first column (cells one to five in Figure 3), the second strip covered the second column (cells six to ten in Figure 3), and so on.

These five strips were then rearranged into a single column by concatenating them vertically. This new column was then appended to those generated from the previous frames using the same procedure. The overall process is illustrated in Figure 3.

5 Implementing Binary Logic Gates

Once the platform was able to transmit BZ oscillations between cells, such as between an activated cell being stirred and an inactive cell where the stirrer was not rotating (Figure 2c), the next objective was to use this functionality to implement binary logic gates. To build linearly separable binary logic gates, such as the "AND" and "OR" gates, we focused on the fact that BZ excitation waves generated at an activated cell easily propagate to direct neighbours (i.e., distance = 1), but do not propagate to cells further away (i.e., distance > 1). This phenomena can be seen on Figure 4. This Figure represents a 30 minute experiment, where the stirrers in the fifth column (cells 21 to 25 – see

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

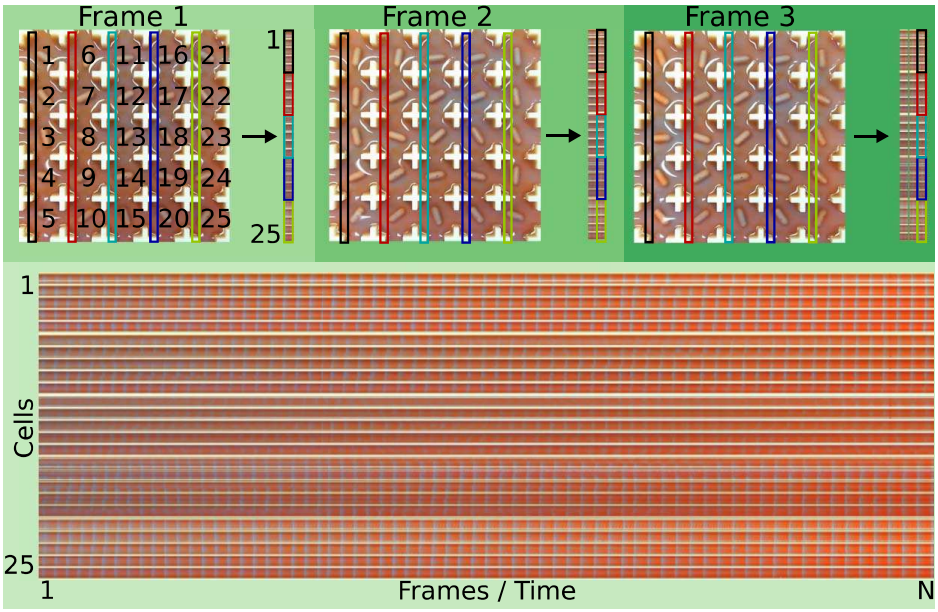


Figure 3. Visualising the experimental videos as 2D timelines. Top: The platform consisted of 25 cells arranged in a 5 × 5 grid. For each frame, a vertical strip with a width of 1 pixel was extracted from each of the five platform columns. These strips were concatenated vertically to form a single column representing the state of the BZ medium at that time point. This procedure was repeated for every frame, with each newly generated column appended to the previously constructed ones. Bottom: Example of this method applied to a 30-minute video. Each blue mark indicates a BZ oscillation occurring within the corresponding cell.

Figure 3 for reference) were enabled, while all the other stirrers were disabled. As it can be seen, the BZ oscillations are clearly visible in the cells where the stirrers were enabled. They are also visible in the adjacent fourth column (where the stirrers were disabled), but as we move towards the other columns, the BZ oscillations are barely visible.

The spatial configurations used to implement the "AND" and "OR" gates were identified through a combination of prior knowledge of BZ wave propagation plus exploratory experiments. In particular, it is well established that excitation waves in BZ media propagate locally through reaction–diffusion coupling, with synchronisation and wave transmission being strongly related to the distance between coupled regions (Kuze et al., 2021). Following this, we systematically tested a range of input–output separations within the grid to determine how far oscillations could reliably propagate.

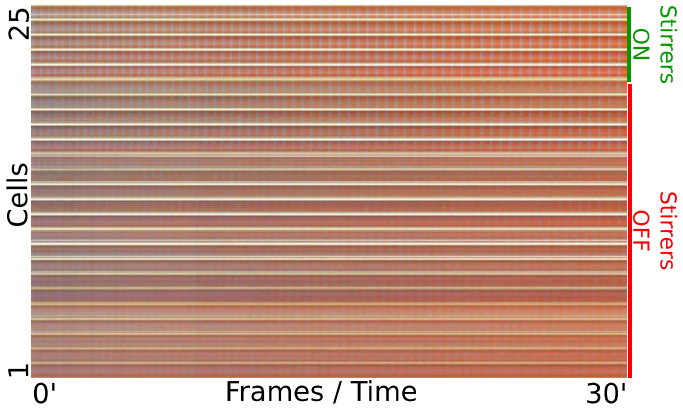


Figure 4. Timeline constructed using the methodology described on Figure 3. In this experiments, cells 21 to 25 (see Figure 3 for reference) had their stirrers enabled, while all the other stirrers were disabled. As it can be seen on the timeline, this mean that the BZ oscillations (blue lines) were clear in the first five rows (cells 21 to 25). They transmitted to the next five rows (cells 16 to 20) where blue lines are also clear, but they are barely visible in the rest of the rows.

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

By exploiting this feature, and based on the observed propagation range of the oscillations, we identified spatial configurations that reproduce 2-bit logic gates using the distance between inputs to implement either an "AND" or a "OR" gate (Figure 5). Here, by input we mean a full column. Thus, since binary logic gates have two inputs, two columns are used as input. When the distance between the two inputs (denoted in Figure 5 as X and Y, respectively) is set to 3 and the output is set as the middle column between them, our platform behaves like an "AND" gate, because oscillations in the output column are only observed when both inputs are activated, and a global oscillation emerges (Figure 5a). On the other hand, when the distance between the inputs is set to 2 and the column between them is defined as output, the system behaves like an "OR" gate, because whenever any of the inputs is enabled, the oscillations propagate to the output column (Figure 5b).

To implement a non-linearly separable logic gate, such as the "XOR" gate, we exploited the fact that when all of the stirrers were activated, globally synchronised wave patterns emerged from the system. We speculate that the emergence of globally synchronised patterns comes from two properties of the BZ reaction: (1) Sustained oscillation: once activated, a BZ reaction tends to sustain its oscillation even after the stirrer is off. Thus, each cell can be considered as an oscillator. (2) Signal convolution: the BZ oscillations generated at a cell propagate to neighbouring cells where other oscillations are generated. The waves collide and interact to form a synchronized oscillation. Thus, the whole BZ grid system can be viewed as a coupled oscillator system, or more in general, as a chemical dynamical system that consistently converts stirrer input patterns into patterns of excitation waves (Liu et al., 2022). This configuration was also identified empirically, based on its ability to generate distinct and reproducible phase patterns for each input combination.

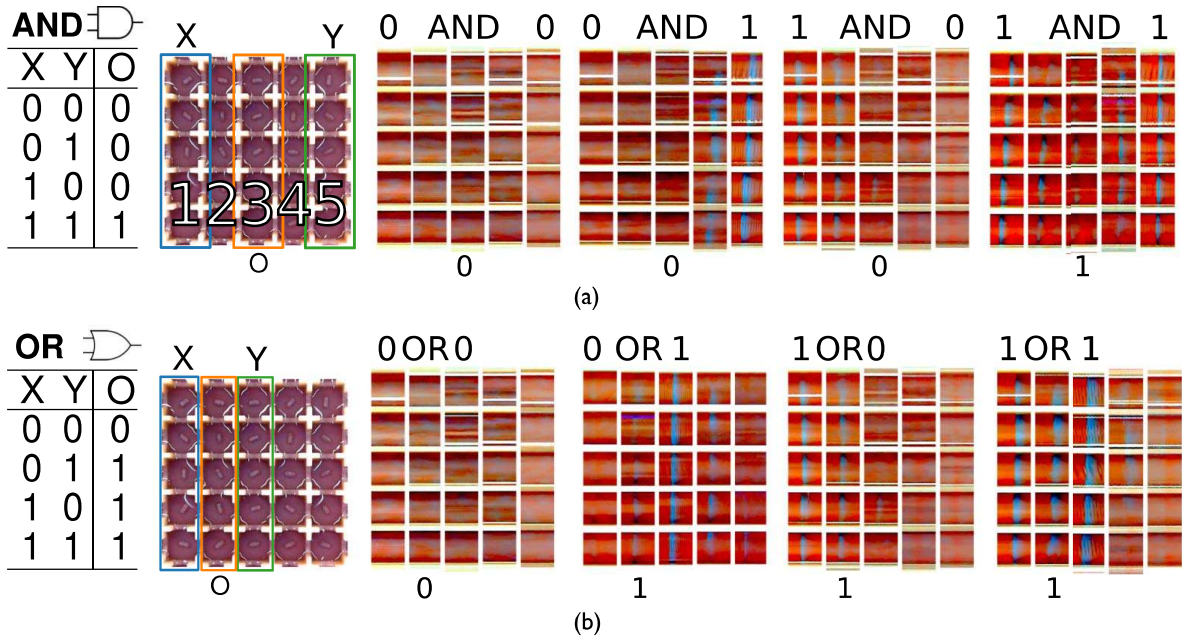


Figure 5. Designing "AND" and "OR" logic gates using the distance between the BZ input oscillations. The leftmost part of this figure describes the truth tables for both the "AND" and "OR" gates. "X" and "Y" are the input bits, while "O" is the output bit. Next to the truth tables, there it can be seen how using the distance between inputs the platform can emulate said logic gates. If X and Y are separated by 3 cells it behaves like an "AND" gate (X = column 1, Y = 5, O = 3), while if they are separated by 1 cell it behaves like an "OR" gate (X = 1, Y = 3, O = 2). The following eight pictures are different snapshots of the platform where it can be seen how the platform actually emulates these logic gates. These pictures do not represent a single frame from the platform, but a sequence of frames taken from a "timeline" as described in Figure 3. That is why the oscillations appear as single blue lines, which means that an oscillation appeared for a number of frames. The base colour (no oscillation) is orange, and a blue line represents an oscillation. The top row represents the "AND" gate, and there it can be seen that the output column only oscillates when both inputs oscillate. The bottom row represents the "OR" gate, and in this case the output column oscillates whenever any of the inputs oscillate.

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

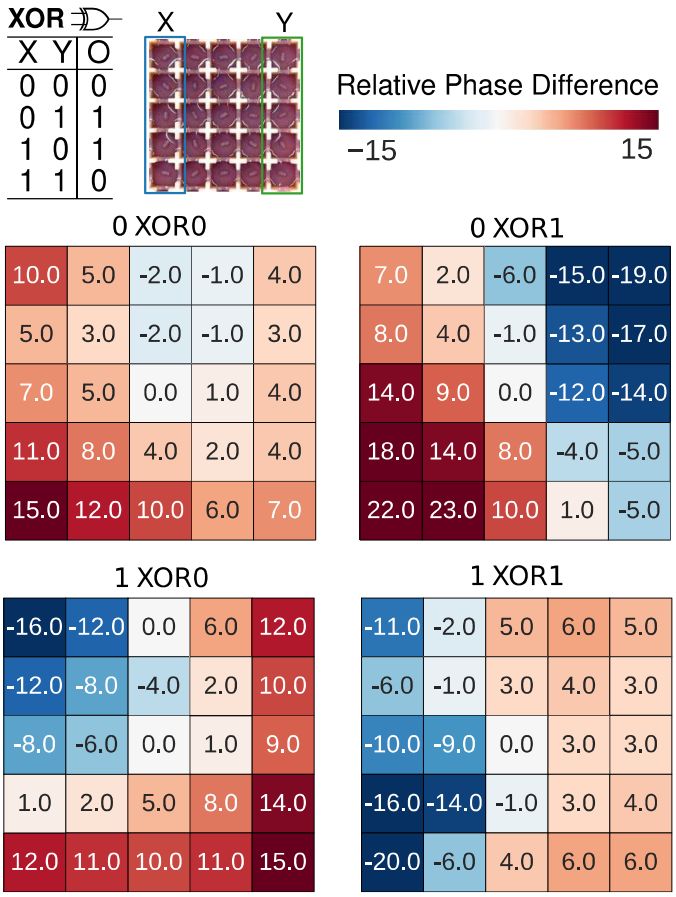


Figure 6. Designing "XOR" logic gates using the phase difference between localised oscillations. The leftmost and rightmost five cells (columns) are considered as inputs (X, Y). The heat maps show the phase differences between the central cell (position 3, 3) and other cells. Each input pattern shows a distinctive BZ excitation wave pattern.

Figure 6 shows how the XOR was implemented by exploiting the BZ oscillations synchronisation between cells. In this case, the ON/OFF binary state of inputs employed two different stirrer speeds (ON for the full speed and OFF for the slowest rotation speed possible), instead of fully activating or disabling stirrers as we did before. In the BZ grid, the leftmost and rightmost five cells were considered as inputs, X and Y, respectively. If only a column of five adjacent stirrers were rotated at full speed (the ON state), the stirrer motion generated a uni-directionally propagating wave pattern. This wave pattern was found to be symmetric, thus both (0, 1) and (1, 0) inputs produced mirrored wave patterns. In contrast, the (0, 0) and (1, 1) inputs resulted in different yet reproducible wave patterns. To quantify these different wave patterns, we calculated the phase differences between the central cell and the other cells by cross correlation analysis (Figure 6). In particular, (0, 1) and (1, 0) patterns show a bipolar pattern where one side ("0" or non-activated side) shows positively shifted phase (indicated in red) while the other side ("1" or activated side) shows negatively shifted phase because it was observed that excitation waves tend to propagate towards the cells with activated stirrers, thus the phase is delayed. On the other hand, the other cases, (0, 0) and (1, 1), showed different non-bipolar patterns. Therefore, the XOR gate can be implemented by defining the bipolar wave pattern as output "1" and the other pattern as "0".

These implementations of the "AND", "OR" and "XOR" gates are intended as proof-of-concept demonstrations of how basic forms of computation can be embodied in the platform, rather than as a general or systematic constructions.

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

6 Using Generative AI to Simulate Our Platform

All the experiments discussed so far were executed in a real platform. This had the advantage of knowing that all the data generated perfectly represented the system, but it came with two drawbacks: (1) The experiments were slow to execute. Each experiment took between 30 to 60 minutes. (2) The financial costs of running the platform, in terms of all the chemicals used, were high; and most of the waste generated was not recyclable. The low throughput of the platform, specifically, limited its capacity to make new discoveries, since based on the timings discussed, the system could only perform eight to ten experiments per day. This severely limited the number of different conditions or regimes we could test to, for example, design new logic gates.

As outlined in the introduction, the BZ reaction is a well-characterised system, with known mathematical and physical models. Nevertheless, our aim here is not to simulate the BZ reaction itself, but to simulate it within our platform, so that it can be used to accelerate the discovery of new forms of computation. To achieve this, we focused on using Generative AI, a branch of AI which has been recently used to simulate a wide range of scientific phenomena (Huang et al., 2024; Mekala et al., 2025). Thus, our objective is to train a Generative AI model that can: (1) Simulate the BZ oscillations on their own; (2) simulate how these oscillations can be modified using external inputs, such as the magnetic stirrers we used, and (3) simulate how the BZ oscillations transmit in an array of neighbouring cells, such as the one defined in our platform.

To train this model we drew on the vasts amount of data we collected while conducting the experiments detailed in the previous sections, as well as the experiments we did in previous research (Cronin et al., 2019). During these experiments, we recorded videos of the BZ medium—such as the frames shown in Figure 1e—while simultaneously logging the motor configurations corresponding to each frame. Based on the data collected, instead of attempting to model the dynamics of the fluids and/or the dynamics of the BZ reaction, we decided to take a different approach and consider this as a "frame prediction" problem. This type of problems are well known in Generative AI, and they currently are an intensive area of research (Höppe et al., 2022; Yan et al., 2021). Our goal here is to train a Generative AI model that, given a video of our BZ platform together with the corresponding motor configurations, can predict the next video frame (see Figure 7).

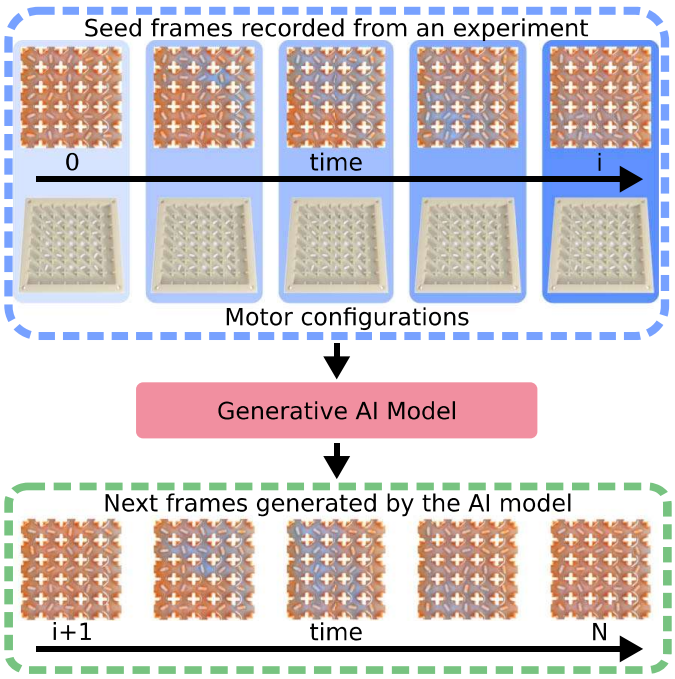


Figure 7. Using Generative AI to predict the next frames. The model takes as input the recorded video frames of the BZ reaction along with the corresponding motor configurations, and predicts the next frame based on this information.

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

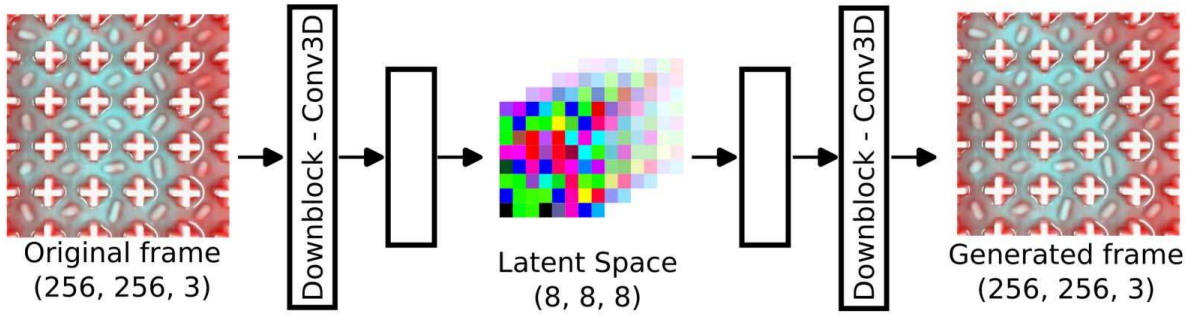


Figure 8. An Autoencoder was used to compress the frames from size (400, 400, 3) to size (8, 8, 8).

To achieve this goal, we focused on Diffusion Models, as these architectures excel at image generation. Among the various Diffusion Models, our implementation is based on the Stable Diffusion architecture (Rombach et al., 2021). This architecture works with latent vectors instead of using full images, which has a big memory and runtime advantage. Given a frame (a full image) we obtained its latent representation using an Autoencoder. Autoencoders are Deep Neural Networks well-known for being very effective at compressing data (Hinton & Salakhutdinov, 2006), and they have been particularly successful at compressing images, which is exactly the type of data we are working with.

Going into the details of our architecture, it first started by resizing the frames from (400, 400, 3) to (256, 256 3) using bilinear interpolation, and then these frames were compressed using an Autoencoder into smaller arrays of shape (8, 8, 8), see Figure 8. Between the encoder and the decoder, defining the latent space, there was a single array "z" (of shape, as discussed, 8, 8, 8). Examples of reconstructed frames generated using this Autoenconder can be seen on Figure 9. We used a single array "z", because this produced more stable results than when following a Variational Autoencoder approach, which uses two arrays. Once this Autoencoder was trained, it was used to create the dataset, which consisted of sequences of frames in latent space extracted from the videos. Every video was divided into overlapping sequences with length "*seq_length*", and then for each frame the Autoencoder was used to obtain their latent representation. Thus, each entry in our dataset was of shape (*seq_length*, 8, 8, 8). We used a sequence length of 128, which provided a good balance between the memory limitations of our hardware and the need to capture the oscillations of the BZ reaction.

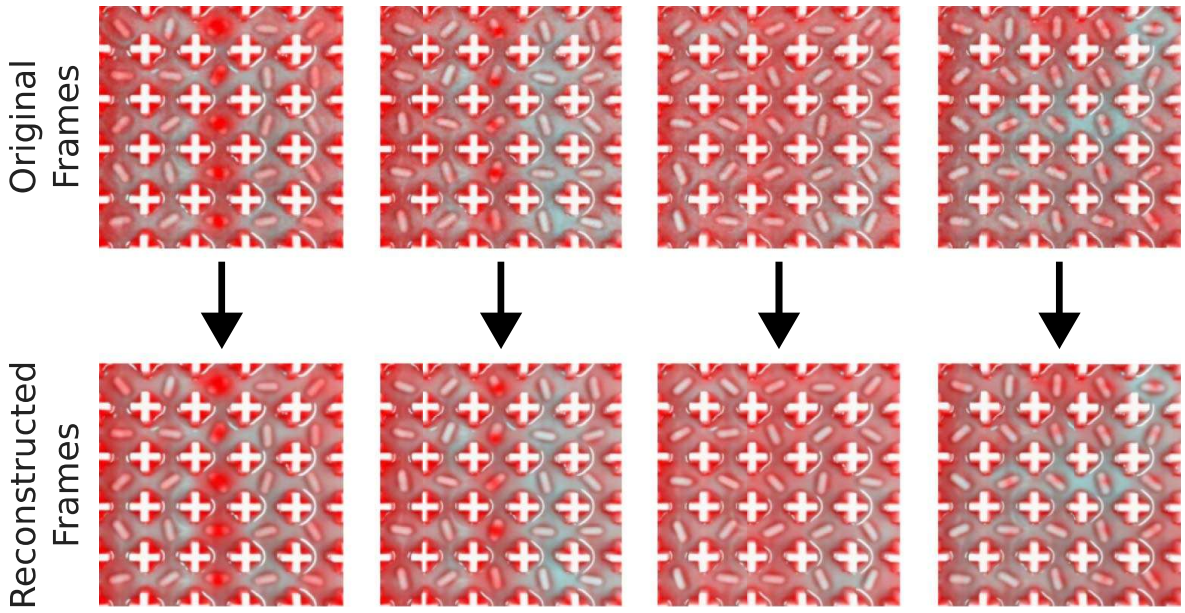


Figure 9. Examples of frames from the original videos, and the reconstructions as outputted by the Autoencoder.

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

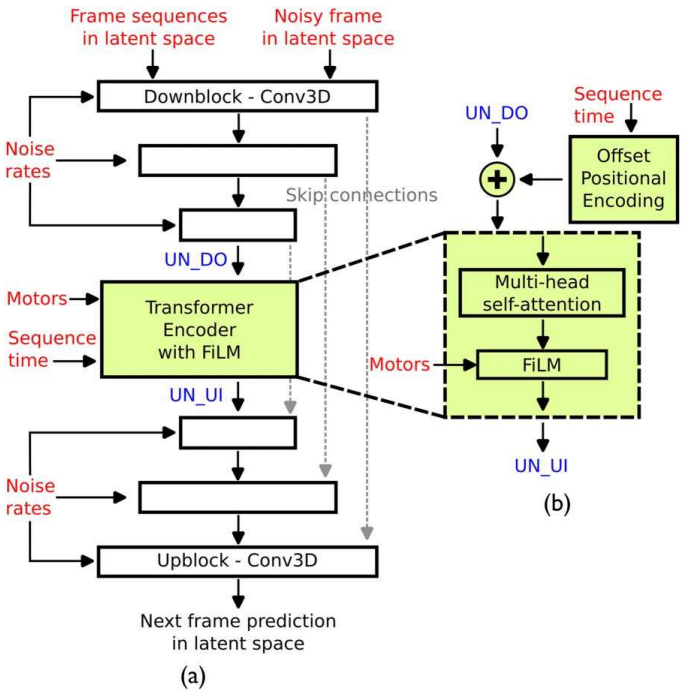


Figure 10. AI Model architecture. In red: Inputs to the model. In blue: Internal inputs. (a) Sequences of compressed frames were inputted into a Diffusion Model, alongside a noisy frame, and this model outputted a de-noised version of this frame, which represented the next frame of the sequence. Our diffusion model used the common U-Net architecture to down-sample and up-sample the data. (b) At the centre of the U-Net, we used Transformer Encoder blocks followed by FiLM (feature-wise linear modulation) layers, to condition the generate of the next frame based on the information from the motors.

These sequences were then used to train our Diffusion Model (see Figure 10a). The model was trained to, given a sequence of frames, predict the next one. To achieve this, the last frame of each input sequence was replaced with noise, and the model learned to remove this noise to reconstruct the original frame. The input sequences were first down-sampled using 3D convolutions, and later on they were up-sampled to reconstruct the data to its original shape. Following the standard U-Net architecture, which is very commonly used in Diffusion Models, we used "skip connections" between the blocks that down-sampled, and the ones that up-sampled. Each block received not only the frame sequences but also the corresponding "noise rates", which indicated the amount of noise added to the last frame. As is common with Diffusion Models, the model was trained to predict the noise that was added initially. Thus, the output of our model was this, a prediction of the noise added during the first step. Once this noise is known, the frame can be reconstructed by subtracting the noise from the original noisy input.

A key feature of the BZ reaction is its temporal oscillations, which must be preserved when generating new data/experiments using the AI model. To improve time-series prediction, we incorporated temporal and positional information into the input sequences. This was achieved by including Transformer Encoder blocks between the down-sampling and up-sampling stages of the U-Net. These blocks, through their Attention mechanisms, are particularly effective at capturing sequential dependencies. The implementation is shown in Figure 10b.

The Transformer Encoder blocks received three inputs: (1) UN_DO—the output from the U-Net down-sampling stage, (2) Motors—the speed of each of the 25 motors at every frame in the sequence, with shape (*seq_length*, 25), and (3) Sequence time—the position of the input sequence within the full experiment from which it was extracted. The "Sequence time" input was processed through an "Offset Positional Encoding" step, and the resulting representation was added to the input "UN_DO". This step closely follows the way positional information is incorporated into the data in Transformer models before it is passed to the encoder or decoder blocks,

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

but with a small variation. As previously discussed, each of our input sequences contained 128 frames, whereas whole experiments had lengths over 4,000 frames. Consequently, an input sequence could be extracted from any position within those 4,000 frames. This offset was incorporated into the positional encoding, allowing the Encoder blocks to capture how the state of the experiment and the BZ oscillations vary depending on their position within the full experiment. Finally, motor information was integrated into the model through a FiLM (feature-wise linear modulation) layer.

With this architecture, given a input sequence of frames in latent space, a noisy frame (also in latent space), and the corresponding motor actuation data, the model predicted and generated the next frame. The new frame could then be appended to the previous ones, enabling the model to generate subsequent frames iteratively. Repeating this loop allowed us to produce as many frames as desired. We evaluated this process using the test set, and the quality of the generated frames was high, making it difficult to discern when the AI began generating them. A common issue in video generation is the progressive degradation of frame quality over longer sequences. In our case, however, significant degradation was not visually observed. We believe the absence of degradation in our predictions comes from the fact that our dataset is highly constrained to this specific problem, resulting in a latent space that was highly specialised for this particular type of experiment. An example of a video generated with our model is shown in Figure 11. As discussed, 128 frames (just over 6 seconds) were used as the seed, after which the AI generated 500 frames (25 seconds). Another known issue with Generate AI models is that they might hallucinate (Rathkopf, 2026). To avoid hallucinations, we used a method known as "retrieval-augmented generation" (Béchard & Ayala, 2024), where the model is constrained to generate data that is only present in a known dataset. In our case, first we created a dataset containing all the frames from all our videos, and we stored them in their latent vector form using the Autoencoder. Then, once the model generated a new latent vector, we searched for the nearest one among the ones in the dataset, and we interpolated between the two.

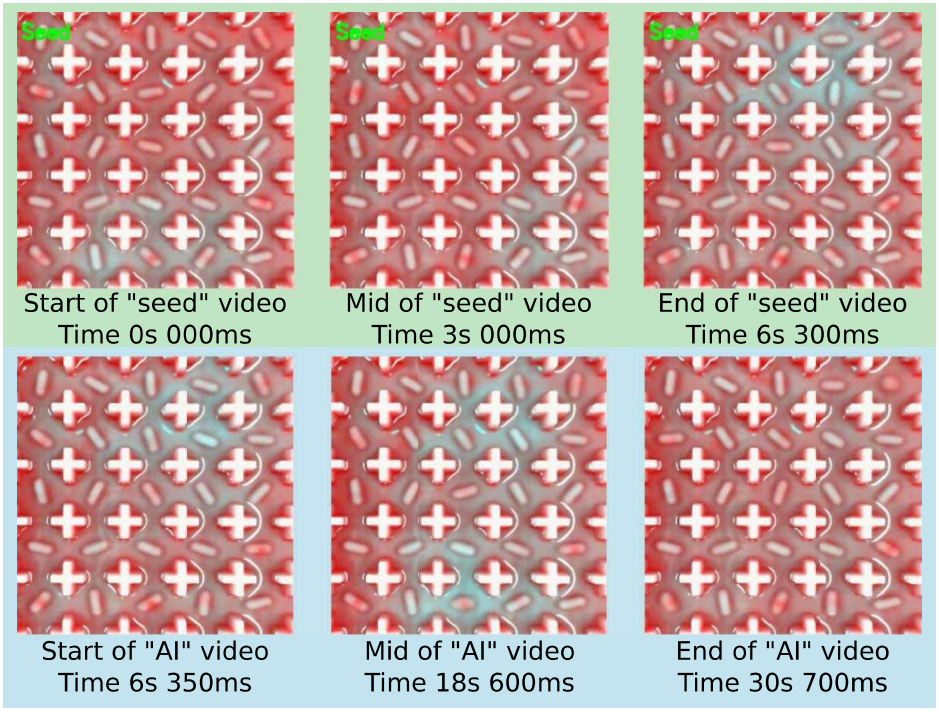


Figure 11. Example of a video generated using our model. Top left (in green background): Frames from an original video, which was used as seed (or prompt) alongside its motor pattern. A total of 128 frames were used as seed, which at 20 FPS equates to just over 6 second of video. Bottom right (in blue background): Frames from a video generated using our model. To generate this video, the frames in green were used as the input sequence.

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

7 Using the Model to Investigate the BZ Reaction *In-Silico*

In the previous section, we demonstrated how Generative AI can be used to model our platform. The purpose of developing this model was to increase experimental throughput: while a single experiment on the physical platform required around one hour, the AI model could potentially generate comparable results within seconds.

To demonstrate the usefulness of our model and its increased data throughput, we paired it with an optimisation algorithm (a Genetic Algorithm or GA) to investigate how different motor patterns influence the behaviours described by the BZ medium. In particular, our goal was to use an optimisation algorithm to find motor patterns that either maximised or minimised a user-defined behaviour. A key property of the BZ reaction is that it oscillates, therefore, the fitness function for our optimisation process was defined as the generation of the strongest possible oscillations. By default the BZ reaction has a red-ish colour, and when an oscillation happens it shifts to blue before eventually returning to red. This phenomenon could be seen in many of the previous figures, such as in Figure 11. In this figure the very last frame (bottom right) has no oscillations (all cells are red), while the previous one (bottom left) has areas with blue cells.

To implement this fitness function, given a video where all the frames were represented in RGB, we extracted the red and blue channels, and we analysed the red–blue difference signal over time for each experiment. The fitness score was calculated based on how many times the signal oscillated and how strong those oscillations were. To build the signal, for each frame (represented in RGB) we took the average red channel, minus the average blue channel. This produced a signal that went up when the frame was more red, and down when it was more blue. This signal was also smoothed to reduce noise. Every time the smoothed signal crossed zero (changed sign), it meant the video switched from red-dominant to blue-dominant (or vice versa). The number of these zero-crossings was the number of oscillations. For each oscillation (between two zero crossings), we measured the peak-to-peak amplitude (how far the signal went up and down), and calculated the average amplitude for all the crossings. Very small oscillations under a set threshold were ignored as noise. The final score was the number of oscillations multiplied by the average amplitude.

This fitness function (F) is then described as:

$$F = N \cdot \bar{A} \tag{1}$$

where

- N = number of zero-crossings in the smoothed signal (oscillations),
- $\bar{A}$ = average peak-to-peak amplitude of valid oscillations.

Formally,

$$\bar{A} = \frac{1}{M}\sum_{i=1}^{M} A_i, \tag{2}$$

with

$$A_i = \max_{t\in[z_{i-1},z_i]} s(t) - \min_{t\in[z_{i-1},z_i]} s(t), \tag{3}$$

subject to the condition

$$A_i \geq A_{\min}, \tag{4}$$

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

where:

- $s(t)$ is the smoothed red–blue difference signal,
- z_i are the zero-crossing indices,
- M is the number of oscillations that satisfy the amplitude threshold,
- and $A_{\min}$ is the minimum amplitude considered significant.

This fitness function was used in conjunction with a GA to identify motor configurations that produced strong oscillations, corresponding to high fitness scores.

The GA was setup following a standard implementation with the following parameters:

- **Genes:** Our platform contained 25 individually controllable motors. In the genetic representation, each motor was associated with a single gene that encoded its rotation speed, resulting in a total of 25 genes per individual.
- **Population size:** We used a population of 160 individuals, the maximum our hardware could handle.
- **Initial population:** Individuals were initialised randomly. Each gene was set using `numpy`'s `random` function minus 0.5, giving initial gene values between −0.5 and 0.5.
- **Elite size:** The elite size was set to 50, corresponding to one-third of the population.
- **Selection:** Generational selection was performed using the "Roulette wheel" algorithm.
- **Mutation:** Each gene underwent mutation with a probability of 0.05%.
- **Crossover:** To breed new individuals, two parents were selected via the Roulette algorithm. A random crossover point R between 0 and 24 was chosen, with the offspring inheriting genes 1 through R from the first parent and genes $R + 1$ through 25 from the second parent.
- **Generations:** A total of 50 generations were evolved, which took approximately two days on our hardware.

Once a population for a given generation was assembled, the next step was to evaluate each individual and compute its fitness score. For each individual, the corresponding experiment was generated from its genes, with each gene directly specifying the speed of one motor in the BZ platform. This motor pattern was inputted into the Generative AI model described earlier, which then produced a video simulation of the experiment. Each video consisted of 500 frames, a process that took approximately 30 minutes. However, since all individuals from the same generation could be generated in parallel, all 160 experiments could be completed within the same 30-minute time-frame. After generating the videos, the fitness score for each individual was calculated using the previously defined fitness function. Finally, the best-performing individuals were selected via the "Roulette wheel" method, and these individuals were bred and mutated to produce the next generation.

Figure 12 shows an example of a complete GA run, from generation 0 to 49 (50 generations in total). In this GA run, the fitness function was calculated for the middle cell only, located at position (3, 3), making the objective to identify motor configurations that generated strong oscillations in this specific cell. Figure 12a shows the evolution of fitness scores over the generations. In generation 0, the average score was close to 0, while by the final generation it approached 200. The best individual appeared in generation 28, achieving a score just over 800. The overall upward trend in fitness scores demonstrates that the GA succeeded in optimising the fitness function by exploring different motor configurations.

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

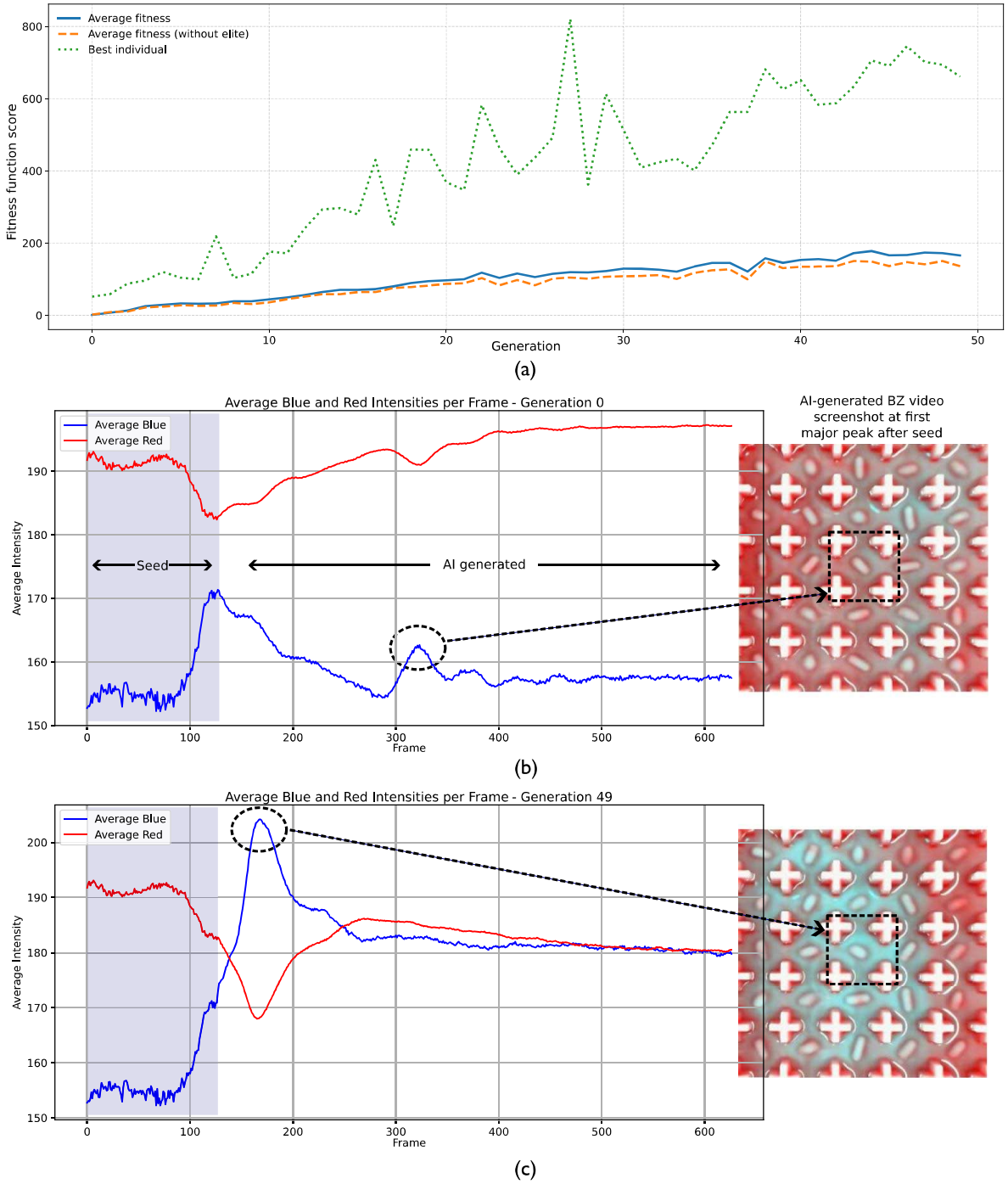


Figure 12. Optimising the motor patterns using a GA to obtain strong BZ oscillations in the middle cell. (a) Plot showing the evolution of the fitness scores from the first generation to the last one. (b) Evolution of the red-blue signals for an individual from the first generation. The blue shaded area represents the seed frames which were fetched directly from the test dataset, while the rest of the frames were generated using our AI model. For each frame, the red line shows the average value of the red channel (from RGB), while the blue line shows the average value of blue channel. To the right of this plot, a screenshot from a blue peak can be seen. In this experiment, the red and blue signals never crossed, meaning the fitness score for this individual was 0. (c) Evolution of the red-blue signals for an individual from the last generation. In this experiment, a strong oscillation can be seen around the 180th frame. Based on the fitness function as described before, the fitness score for this individual was 199.25.

Figure 12b shows the evolution of the fitness score for an individual from the first generation. The plot shows the red and blue channel values over the frames of the video. The initial section, highlighted with a shaded blue background, represents the "seed" frames—portions of the video directly taken from the test set (i.e., from a real experiment). The remaining frames were generated

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

Table 1. Example results from five different optimisation runs using a Genetic Algorithm.

GA Run	1st Gen Avg Score	49th Gen Avg Score	Best Score
1	21.2	243.0	449.1
2	0.95	94.6	358.8
3	1.23	165.6	745.3
4	6.28	219.5	313.5
5	51.3	199.2	606.8

by the AI model described previously. In this example, the red and blue signals never crossed, indicating that no oscillations occurred; as a result, this individual received a fitness score of 0. To the right of the plot, a screenshot shows the highest blue peak within the AI-generated section.

Figure 12c shows the evolution of the fitness score for an individual from the last generation. In this case, the red and blue signals crossed, indicating that an oscillation occurred in the centre cell. Based on the fitness function described earlier, this individual received a fitness score of 199.25.

Similar GA runs were repeated to reproduce the optimisation results, which are summarised in Table 1. In each run, the sequence of frames used as the "seed" for the Generative AI model was different and randomly chosen. Similarly, the initial motor pattern during the "seed" portion varied for each run, corresponding to the motor pattern associated with the selected seed sequence.

8 Benchmarking the AI Model

To benchmark the AI model, 128 sequences were selected randomly from the test set. These sequences contained 256 steps. The first 128 steps were used as input (seed) to the AI model, while the following 128 steps were used to compare the model outputs with the corresponding ground-truth data from the test set. To perform this comparison, as in the previous section, we used the centre cell as the target for the calculations. Therefore, the comparison focused on the difference in the blue channel of the platform's centre cell between the test set and the AI-generated data.

In Figure 13a, the black lines show the difference in blue-channel value at the centre cell for each of the 128 tested sequences. While the y-axis is scaled between -20 and 40, the theoretical maximum range could go from -256 to 256. The red line represents the mean difference across the 128 sequences at each step, while the shaded red region represents the corresponding standard deviation. To calculate the overall mean and standard deviation we converted the differences to absolute values - so that positive and negative values cannot cancel themselves out. We obtained an overall average difference of 3.68, with an overall standard deviation of 5.06. These results can be observed in this figure, where most of the experiments cluster around the Y-0 position, creating a large shaded black area, with the red line (mean) also located in this region. While these results are encouraging, it is important to remember that the BZ reaction remains at the non-oscillatory state (red colour) most of the time, with brief oscillations (blue colour).

To further explore the benchmark results, we focused on the experiment displaying the largest difference between test and AI-generated data. This experiment is highlighted with the orange line in Figure 13a. Checking this individual experiment from the test-set, it can be observed that this experiment initially displayed oscillatory behaviour (marked as "max" in the plot), while the corresponding AI-generated sequence remained in a non-oscillatory state. Then, after the test data returned to a non-oscillatory state, oscillations emerged in the AI-generated data. In both cases, oscillatory behaviour is observed. However, the oscillations were separated in time by a phase

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

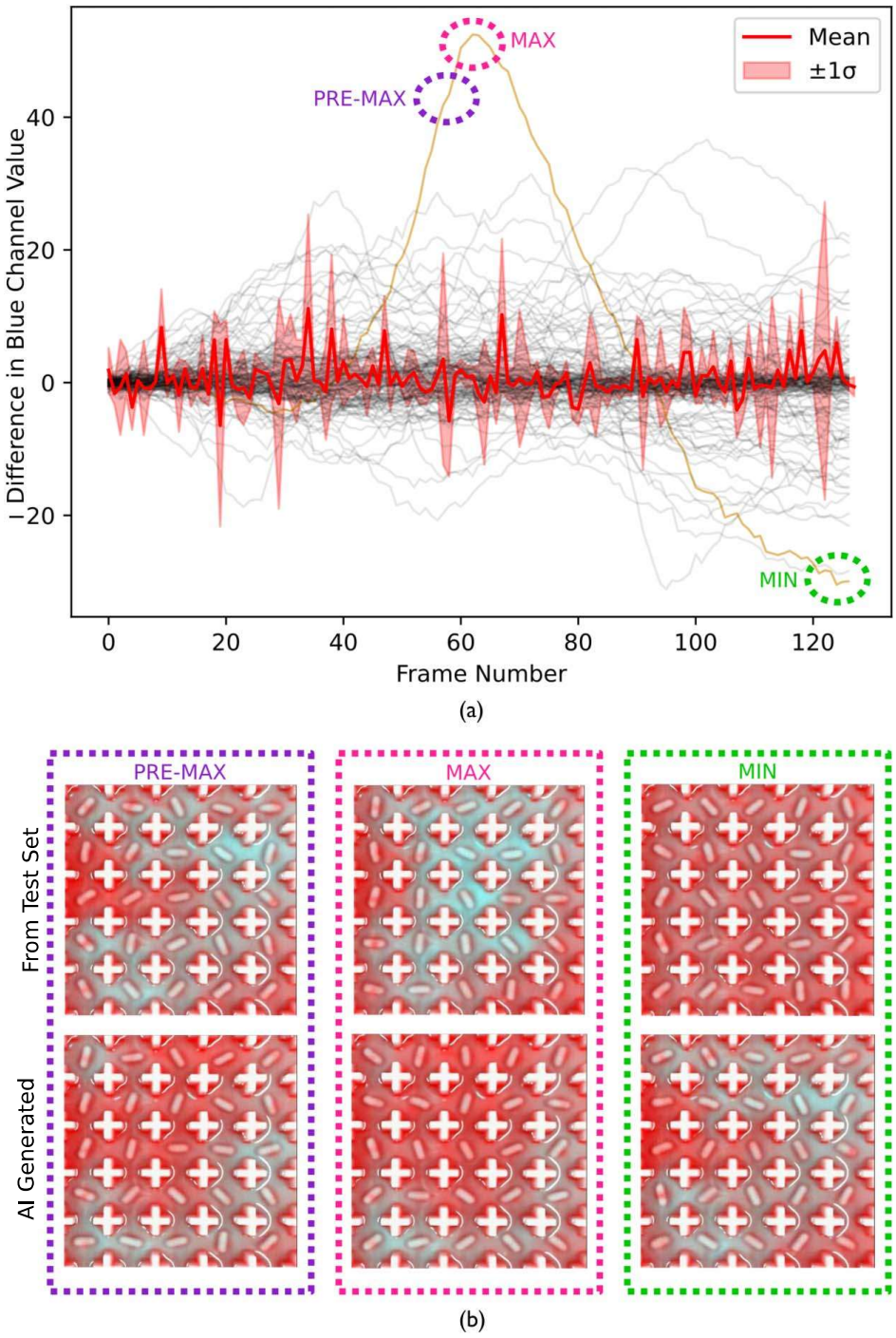


Figure 13. Benchmarking the AI model by comparing sequences in the test set against sequences generated by the model. (a) Plot showing the difference in the blue-channel value at the centre cell for 128 test-set sequences, compared against sequences generated through the AI model using the same seed. Each black line represents the difference between an experiment from the test set, and its corresponding experiment generated using the AI model. While the plot is autoscaled between −20 and 40, the maximum range goes from −256 to 256. The red line represents the mean difference per frame number, while the shaded red area represents the standard deviation. The orange line represents the experiment with the biggest difference, which is further explored in B. (b) Three snapshots comparing one experiment (orange line above) from the test set against the frames generated using the same seed.

difference. Interestingly, as shown in Figure 13b, the snapshot from the test set at the "pre-max" time-point closely resembles the snapshot from the AI-generated data at the "min" time-point.

It is important to remark that our benchmarking tests focused on a single cell located at the centre of the platform, and therefore it did not fully capture all aspects of the spatiotemporal

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

dynamics of the system. In future work, these benchmarking tests could be extended to include all the cells, as well as different input and output patterns.

9 Discussion and Conclusions

While in previous research binary logic gates have been modelled using BZ reaction diffusion systems, here we have shown an example of a chemical processor with real-time input used to experimentally demonstrate binary logic behaviours within the platform. Therefore, our platform builds on prior work that demonstrated Boolean logic in BZ systems, extending it through real-time actuation and integration with a digital control layer.

To reach this result, we first had to show that we could reliably control the oscillations via an external stimulus, and physical stirring of the individual cells using a magnetic stirrer was found to be effective as a control. Oscillating cells could then initiate oscillations in neighbouring cells if they were fluidically connected. Using these properties from our platform, we experimentally demonstrated reproducible "AND", "OR" and "XOR" logic-like behaviours by configuring the spatial arrangement of the inputs and exploiting the resulting oscillatory dynamics. The wave propagation patterns that emerged from these inputs were reproducible between experiments. Nevertheless, the logic gate implementations shown in this research are not intended to be used as a general framework for building a chemical computer, but instead as examples of how the platform could be used to perform simple computations by controlling the spatiotemporal dynamics of the BZ medium.

While in this study we focused on implementing the "AND", "OR" and "XOR" logic gates, a possible route to achieve universal computation would be to implement a "NAND" gate. "NAND" gates are functionally complete, meaning that all the other boolean operations can be implemented with "NAND" combinations, and thus an entire processor can be created using "NAND" gates alone. In this research we did not attempt to create "NAND" gates because we could not create "NOT" operations. The main reason behind this is that in our set-up the chemistry only oscillated ("1" state) when the stirring was enabled, and we did not find a way to induce oscillations with disabled motors. Future research could focus on this aspect, and if "NOT" operations were achieved, this would enable us to also perform "NAND" operations, and thus the platform could be said to be functionally complete.

To further explore our platform, we modelled it using Generative AI. The model was trained using only videos from previous experiments and was able to generate videos of possible future experiments. We then employed this Generative AI model as a surrogate to optimise the platform's behaviour using a Genetic Algorithm. For the optimisation, the fitness function was defined to find configurations that produced strong BZ oscillations. The GA identified motor configurations that, within the surrogate model, maximise the defined fitness function, and the corresponding generated sequences displayed strong oscillatory behaviour.

Nevertheless, these optimisation results were obtained entirely *in-silico* and were not experimentally validated on the physical platform, and that would be an important next step. Moreover, the fitness function we defined was relatively simplistic. Future work could involve defining more sophisticated functions with a closer connection to unconventional computation.

In this study we implemented a hybrid chemical computing system using the BZ medium as processing unit, and a conventional digital computer as control unit. An advantage of the described system is its adaptability because the number of cells used can be easily increased, since only a new 3D-printed vessel is required. Furthermore, the 3D-printed grid itself can be modified by changing, for example, the gap width between two neighbouring cells, or the shape of the cells, which will also affect the wave propagation. As an example of these possible enhancements, we have recently developed a similar platform to the one described here, but containing a 7-by-7 grid (instead of 5-by-5) (Sharma et al., 2024). This newer platform incorporates new stir bars at the interface between cells - added the ones at the centre of each cell as in this research- enabling us to have a finer control of how BZ oscillations transmit between cells. We believe that further investigation and implementation of

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

chemical computing systems will pave the way to exploit the rich dynamic of natural unconventional computing substrates.

Data Availability

The data supporting the findings of this study consist of multiple video files with a total size of approximately 1 TB. The curated dataset is available from the corresponding author upon reasonable request.

Code Availability

The source code is available at https://github.com/thephet/BZ1gen/.

Acknowledgments

L.C. gratefully acknowledge financial support from the EPSRC (Grant Nos. EP/H024107/1, EP/I033459/1, EP/J00135X/1, EP/J015156/1, EP/K021966/1, EP/K023004/1, EP/K038885/1, EP/L015668/1, EP/L023652/1), the ERC (project 670467 SMART-POM), and the DARPA molecular informatics project. J.M.P.G. acknowledges financial support from the Royal Society with the grant code APX/R1/231098 APEX Awards 2023.

References

Adamatzky, A. (2015). Binary full adder, made of fusion gates, in a subexcitable Belousov-Zhabotinsky system. *Physical Review E*, *92*(3), 032811. https://doi.org/10.1103/PhysRevE.92.032811, PubMed: 26465532

Adamatzky, A. (2017). Fredkin and Toffoli Gates implemented in oregonator model of Belousov-Zhabotinsky medium. *International Journal of Bifurcation and Chaos*, *27*. https://doi.org/10.1142/S0218127417500419

Adamatzky, A., de Lacy Costello, B., & Asai, T. (2005). *Reaction-diffusion computers*. Elsevier Inc. https://doi.org/10.1016/B978-0-444-52042-5.X5000-2

Adamatzky, A., & de Lacy Costello, B. (2002). Collision-free path planning in the Belousov-Zhabotinsky medium assisted by a cellular automaton. *Naturwissenschaften*, *89*(10), 474–478. https://doi.org/10.1007/s00114-002-0363-6, PubMed: 12384724

Arnold, J., Schäfer, F., Edelman, A., & Bruder, C. (2024). Mapping out phase diagrams with generative classifiers. *Physical Review Letters*, *132*(20), 207301. https://doi.org/10.1103/PhysRevLett.132.207301, PubMed: 38829098

Baltussen, M. G., de Jong, T. J., Duez, Q., Robinson, W. E., & Huck, W. T. (2024). Chemical reservoir computation in a self-organizing reaction network. *Nature*, *631*(8021), 549–555. https://doi.org/10.1038/s41586-024-07567-x, PubMed: 38926572

Baron, R., Lioubashevski, O., Katz, E., Niazov, T., & Willner, I. (2006). Logic gates and elementary computing by enzymes. *Journal of Physical Chemistry A*, *110*(27), 8548–8553. https://doi.org/10.1021/jp0568327, PubMed: 16821840

Béchard, P., & Ayala, O. M. (2024). Reducing hallucination in structured outputs via retrieval-augmented generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Volume 6, pp. 228–238). https://doi.org/10.18653/v1/2024.naacl-industry.19

Cachay, S., Henn, B., Watt-Meyer, O., Bretherton, C. S., & Yu, R. (2024). Probabilistic emulation of a global climate model with spherical dyffusion. *NeurIPS*. https://github.com/Rose-STL-Lab/spherical-dyffusion

Conrad, M. (1985). On design principles for a molecular computer. *Communications of the ACM*, *28*(5), 464–480. https://doi.org/10.1145/3532.3533

Cronin, L., Parrilla Gutiérrez, J. M., Tsuda, S., Sharma, A., Cooper, G., Aragon-Camarasa, G., & Donkers, K. (2019). *A programmable chemical computer with memory and pattern recognition*. ChemRxiv. https://doi.org/10.26434/chemrxiv.7712564.v1

Deco, G., & Kringelbach, M. L. (2017). Hierarchy of information processing in the brain: A novel 'intrinsic ignition' framework. *Neuron*, *94*(5), 961–968. https://doi.org/10.1016/j.neuron.2017.03.028, PubMed: 28595052

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

de Lacy Costello, B., & Adamatzky, A. (2005). Experimental implementation of collision-based gates in Belousov-Zhabotinsky medium. *Chaos, Solitons and Fractals*, *25*(3), 535–544. https://doi.org/10.1016/j.chaos.2004.11.056

Dueñas-Díez, M., & Pérez-Mercader, J. (2019). How chemistry computes: Language recognition by non-biochemical chemical automata. From finite automata to turing machines. *iScience*, *19*, 514–526. https://doi.org/10.1016/j.isci.2019.08.007, PubMed: 31442667

Fletcher, W., Klinge, T. H., Lathrop, J. I., Nye, D. A., & Rayman, M. (2025). Real-time computing and robust memory with deterministic chemical reaction networks. *Natural Computing*, *24*, 383–397. https://doi.org/10.1007/s11047-024-09994-1

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature*, *403*, 339–342. https://doi.org/10.1038/35002131, PubMed: 10659857

Gorecki, J., Gizynski, K., Guzowski, J., Gorecka, J. N., Garstecki, P., Gruenert, G., & Dittrich, P. (2015). Chemical computing with reaction-diffusion processes. *Philosophical Tranactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *373*(2046), 20140219. https://doi.org/10.1098/rsta.2014.0219, PubMed: 26078345

Guo, S. Y., Friederich, P., Cao, Y., Wu, T. C., Forman, C. J., Mendoza, D., Degroote, M., Cavell, A., Krasecki, V., Hickman, R. J., Sharma, A., Cronin, L., Gianneschi, N., Goldsmith, R. H., & Aspuru-Guzik, A. (2021). A molecular computing approach to solving optimization problems via programmable microdroplet arrays. *Matter*, *4*(4), 1107–1124. https://doi.org/10.1016/j.matt.2021.03.002

Guzowski, J., Gizynski, K., Gorecki, J., & Garstecki, P. (2016). Microfluidic platform for reproducible self-assembly of chemically communicating droplet networks with predesigned number and type of the communicating compartments. *Lab on a Chip*, *16*(4), 764–772. https://doi.org/10.1039/C5LC01526J, PubMed: 26785761

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507. https://doi.org/10.1126/science.1127647, PubMed: 16873662

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *NeurIPS*. https://github.com/hojonathanho/diffusion

Holley, J., Jahan, I., De Lacy Costello, B., Bull, L., & Adamatzky, A. (2011). Logical and arithmetic circuits in Belousov-Zhabotinsky encapsulated disks. *Physical Review E*, *84*, 056110. https://doi.org/10.1103/PhysRevE.84.056110, PubMed: 22181476

Höppe, T., Stockholm, K., Mehrjou, A., Bauer, S., Nielsen, D., & Dittadi, A. (2022). Diffusion models for video prediction and infilling. *Transactions on Machine Learning Research*. https://sites.google.com/view/video-diffusion-prediction

Hsu, T. J., Mou, C. Y., & Lee, D. J. (1994). Effects of macromixing on the oregonator model of the Belousov—Zhabotinsky reaction in a stirred reactor. *Chemical Engineering Science*, *49*(24), 5291–5305. https://doi.org/10.1016/0009-2509(94)00278-9

Huang, T., Zhang, H., Zeng, Y., Zhang, Z., Li, H., Zuo, W., & Lau, R. W. H. (2024). *Dreamphysics: Learning physics-based 3d dynamics with video diffusion priors*. ArXiv. https://doi.org/10.48550/arXiv.2406.01476

Joung, J. F., Fong, M. H., Casetti, N., Liles, J. P., Dassanayake, N. S., & Coley, C. W. (2025). Electron flow matching for generative reaction mechanism prediction. *Nature*, *645*(8079), 115–123. https://doi.org/10.1038/s41586-025-09426-9, PubMed: 40836082

Karimov, A., Kopets, E., Karimov, T., Almjasheva, O., Arlyapov, V., & Butusov, D. (2023). Empirically developed model of the stirring-controlled Belousov–Zhabotinsky reaction. *Chaos, Solitons and Fractals*, *176*, 114149. https://doi.org/10.1016/j.chaos.2023.114149

Kendon, V., Sebald, A., & Stepney, S. (2015). Heterotic computing: Exploiting hybrid computational devices. *Philosophical Transactions of the Royal Society A*, *373*(2046), 20150091. https://doi.org/10.1098/rsta.2015.0091, PubMed: 26078351

Krasecki, V. K., Sharma, A., Cavell, A. C., Forman, C., Guo, S. Y., Jensen, E. T., Smith, M. A., Czerwinski, R., Friederich, P., Hickman, R. J., Gianneschi, N., Aspuru-Guzik, A., Cronin, L., & Goldsmith, R. H. (2023). The role of experimental noise in a hybrid classical-molecular computer to solve combinatorial optimization problems. *ACS Central Science*, *9*(7), 1453–1465. https://doi.org/10.1021/acscentsci.3c00515, PubMed: 37521801

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

Kuze, M., Kitahata, H., & Nakata, S. (2021). Traveling waves propagating through coupled microbeads in the Belousov-Zhabotinsky reaction. *Physical Chemistry Chemical Physics*, *23*(42), 24175–24179. https://doi.org/10.1039/d1cp03916d, PubMed: 34673865

Liu, Y., Pérez-Mercader, J., & Kiss, I. Z. (2022). Synchronization of Belousov-Zhabotinsky oscillators with electrochemical coupling in a spontaneous process. *Chaos*, *32*(9), 093128. https://doi.org/10.1063/5.0096689, PubMed: 36182363

McEvoy, M. A., & Correll, N. (2015). Materials that couple sensing, actuation, computation, and communication. *Science*, *347*(6228), 1261689. https://doi.org/10.1126/science.1261689, PubMed: 25792332

Mekala, R. R., Garratt, E., Muehle, M., Srinivasan, A., Porter, A., & Lindvall, M. (2025). AI-guided frame prediction techniques to model single crystal diamond growth. *Journal of Vacuum Science & Technology A*, *43*, 032708. https://doi.org/10.1116/6.0004290

Park, H., Onwuli, A., & Walsh, A. (2025). Exploration of crystal chemical space using text-guided generative artificial intelligence. *Nature Communications*, *16*(1), 4379. https://doi.org/10.1038/s41467-025-59636-y, PubMed: 40355453

Prokup, A., Hemphill, J., & Deiters, A. (2012). DNA computation: A photochemically controlled and gate. *Journal of the American Chemical Society*, *134*(8), 3810–3815. https://doi.org/10.1021/ja210050s, PubMed: 22239155

Qian, L., & Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, *332*(6034), 1196–1201. https://doi.org/10.1126/science.1200520, PubMed: 21636773

Rambidi, N. G., Maximychev, A. V., & Usatov, A. V. (1994). Molecular image-processing devices based on chemical reaction systems. *Advanced Materials for Optics and Electronics*, *4*(3), 191–201. https://doi.org/10.1002/amo.860040303

Rathkopf, C. (2026). *Hallucination, reliability, and the role of generative AI in science*. ArXiv. https://doi.org/10.48550/arXiv.2504.08526

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). *High-resolution image synthesis with latent diffusion models*. ArXiv. https://doi.org/10.48550/arXiv.2112.10752

Sharma, A., Ng, M. T.-K., Gutierrez, J. M. P., Jiang, Y., & Cronin, L. (2024). A programmable hybrid digital chemical information processor based on the Belousov-Zhabotinsky reaction. *Nature Communications*, *15*(1), 1984. https://doi.org/10.1038/s41467-024-45896-7, PubMed: 38443339

Sourjik, V. (2004). Receptor clustering and signal processing in E. coli chemotaxis. *Trends in Microbiology*, *12*(12), 569–576. https://doi.org/10.1016/j.tim.2004.10.003, PubMed: 15539117

Stevens, W. M., Adamatzky, A., Jahan, I., & de Lacy Costello, B. (2012). Time-dependent wave selection for information processing in excitable media. *Physical Review E*, *85*, 066129. https://doi.org/10.1103/PhysRevE.85.066129, PubMed: 23005184

Sun, M.-Z., & Zhao, X. (2013). Multi-bit binary decoder based on Belousov-Zhabotinsky reaction. *Journal of Chemical Physics*, *138*(11), 114106. https://doi.org/10.1063/1.4794995, PubMed: 23534626

Tóth, Á., & Showalter, K. (1995). Logic gates in excitable media. *Journal of Chemical Physics*, *103*, 2058–2066. https://doi.org/10.1063/1.469732

Toth, R., Stone, C., Adamatzky, A., de Lacy Costello, B., & Bull, L. (2008). Dynamic control and information processing in the Belousov-Zhabotinsky reaction using a coevolutionary algorithm. *Journal of Chemical Physics*, *129*(18). https://doi.org/10.1063/1.2932252, PubMed: 184708, PubMed: 19045424

Toth, R., Stone, C., Adamatzky, A., de Lacy Costello, B., & Bull, L. (2009). Experimental validation of binary collisions between wave fragments in the photosensitive Belousov-Zhabotinsky reaction. *Chaos, Solitons and Fractals*, *41*(4), 1605–1615. https://doi.org/10.1016/j.chaos.2008.07.001

Tsompanas, M.-A., Fullarton, C., & Adamatzky, A. (2019). Belousov-Zhabotinsky liquid marbles in robot control. *Sensors and Actuators B: Chemical*, *295*, 194–203. https://doi.org/10.1016/j.snb.2019.04.152

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *31*, 5999–6009.

Yan, W., Zhang, Y., Abbeel, P., & Srinivas, A. (2021). *Videogpt: Video generation using vq-vae and transformers*. ArXiv. https://doi.org/10.48550/arXiv.2104.10157

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026

Yang, S., Bögels, B. W. A., Wang, F., Xu, C., Dou, H., Mann, S., Fan, C., & de Greef, T. F. A. (2024). DNA as a universal chemical substrate for computing and data storage. *Nature Reviews Chemistry*, *8*(3), 179–194. https://doi.org/10.1038/s41570-024-00576-4, PubMed: 38337008

Zhu, S.-L., & Wang, Z. D. (2003). Unconventional geometric quantum computation. *Physical Review Letters*, *91*(18), 187902. https://doi.org/10.1103/PhysRevLett.91.187902, PubMed: 14611315

Downloaded from direct.mit.edu/artl/article-pdf/32/1/99/2622028/artl.a.460.pdf by University of Glasgow user on 06 October 2026