https://doi.org/10.1038/s44335-024-00016-y

Hyper flexible neural networks rapidly switch between logic operations in a compact four neuron circuit

Check for updates

Alexander James White[1,2,7] ✉, Belle Liu[1,3,7], Ming-Ju Hsieh[1,3], Meng-Fan Chang[4,5], Kuo-An Wu[3] & Chung-Chuan Lo[1,6] ✉

Biological neural circuits at various levels exhibit rapid adaptability to diverse environmental stimuli. Such fast response times imply that adaptation cannot rely solely on synaptic plasticity, which operates on a much slower timescale. Instead, circuits must be inherently hyper-flexible and receptive to switches in functionalities without changes in network structure. This biological flexibility is a fruitful mechanism for constructing artificial reconfigurable circuits, whether they are spiking or non-spiking. In this study, we demonstrate that a four-neuron circuit can rapidly and controllably switch between 24 unique logical functions while maintaining the same set of synaptic weights. Moreover, we show that this reconfigurability works for several different underlying neuronal architectures and strikingly can be applied to a network composed of any sigmoid-shaped activation function. We conclude with proof-of-concept applications showing that we can perform standard tasks, such as a full-adder, as well as event-based conditional computing, such as detecting unexpected motion.

Across all organisms, nervous systems have been capable of dealing with the vast variability in their environment. Given the potentially rapid and unpredictable nature of natural stimuli, there is often no time to reconfigure a nervous system's synaptic connections. To address this issue, environmental stimuli or contextual stimuli push the system into a different dynamical mode, that allows it to perform a different set of computations[1–3]. Likewise, there has been a recent push for reconfigurable computing, especially in the neuromorphic space, in which a circuit can be reprogrammed without changing its connectivity[4]. Biological network's inherent rapid flexibility can be leveraged for designing reconfigurable circuits, be it neuromorphic[5,6], organic[7,8], or electronic[4]. Using direct inspiration from biological flexibility, we can create reconfigurable neuromorphic circuits[9].

Many studies have demonstrated this flexibility, for instance, studies have shown how small neural networks in crabs and aplysia are able to switch their function in response to different environment stimuli[10–21]. Furthermore, cortical studies have shown how contextual cues can cause the networks to change activity patterns[2,11,16,22–26]. Collectively, these phenomena of performing different behaviors based on contextual or environmental information is known as flexibility[1,13].

Interestingly, this flexibility is important beyond just choosing the right behavior. There is evidence that flexible responses occur on a microscale[25,27]. That is, contextual information from elsewhere in the brain will cause different responses in small collections of neurons[3,23–29]. This flexibility extends beyond just choosing the right behavior, and can also be used in information gating[3,16,30–32], motor planning[10,14,25,33], and influencing decision networks[13,16,25,30–32]. Moreover, there is evidence that flexibility can exist on a microscale - meaning that contextual information can elicit different responses in small collections of neurons, not just in large-scaled networks[1,3,23,25,27,29,30].

This mechanism is noticeably absent in artificial neural networks, especially deep and convolutional neural networks (DNN and CNN). Such networks perform a single, designated task exceptionally well, but cannot generalize, as the solution hinges on a set of learned synaptic weights[34–38]. If the task changes, so must the weights[36,37,39–41]. While animals do indeed use this strategy to learn tasks, the lack of flexibility in DNNs indicates that synaptic weights alone do not explain the whole story. Many organisms are able to respond to a diverse array of situations without having to relearn a new set of synaptic weights[14,23,27,42]. This is most clear in the motor cortex, where synaptic weights don't change from task to task[10,14,33,38].

Previous work on reconfigurable networks tries to endow networks with increased flexibility. Many circuits make use of memristors[4,6,43,44] and homojunctions[45] to construct reconfigurable

[1]Institute of Systems Neuroscience, National Tsing Hua University, Hsinchu, Taiwan. [2]International Intercollegiate Ph.D. Program, National Tsing Hua University, Hsinchu, Taiwan. [3]Department of Physics, National Tsing Hua University, Hsinchu, Taiwan. [4]Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan. [5]Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan. [6]Brain Research Center, National Tsing Hua University, Hsinchu, Taiwan. [7]These authors contributed equally: Alexander James White, Belle Liu. ✉e-mail: ajw@lolab-nthu.org; cclo@mx.nthu.edu.tw

networks. These reconfigurable circuits can even be used in neuromorphic networks to implement short-time dependent plasticity and synaptic learning[4,45,46]. Moreover, this work has also shown that neurons are complex dynamic systems that are capable of solving a plethora of tasks, especially as a network with synaptic learning[4–6,44,45]. Still, none of these networks seem to make use of biology's control of the underlying dynamics of the network without (or in addition to) changing the synaptic weights[1,14,47,48].

It is suspected that flexibility in the nervous system stems from its highly recurrent nature. Furthermore, these networks contain bifurcations, a qualitative change in the behavior of the system as a parameter, such as contextual input, is varied, leading to the emergence of new dynamic states[49]. Moreover, it is proposed that this is a consequence of the network being near multiple bifurcations, i.e., at a position in which state changes are easy to induce[9,33,50–53]. Being near the cusp of state changes allows the network to rapidly switch to qualitatively different dynamics upon manipulation of certain control parameters[5,33,49]. However, while several studies have attempted to understand how bifurcations aid flexibility[9,13,54], it is still unclear how the nervous system can control networks near such complicated bifurcation structures without altering synaptic weights.

In our previous work, we identified a hyper-flexible four-neuron circuit, named CRIREL (coupled recurrent inhibitory and recurrent excitatory loops). It is composed of two sets of strongly recurrently coupled neurons: a recurrently connected excitatory pair and a recurrently connected inhibitory pair, where these pairs are then weakly coupled with one another. We showed that CRIRELs are flexible because we can control an underlying double-cusp bifurcation present in the network[9,48,53], which allows for new stable or unstable states to emerge as control parameters (in this case, synaptic weights) are varied. However, synaptic weights are not the only controllable parameters, and here we demonstrate how varying bias currents (baseline activity level, implemented as a constant background input current) and input, while keeping synaptic weights fixed, also induces bifurcations and results in flexibility. Therefore, through a detailed study of the dynamics of this microcircuit, we propose a template for how rapid flexibility can be present in neural networks without evoking slow timescale adaptation mechanisms.

In order to systematically investigate the repertoire of functionalities within CRIREL, we classify its output in terms of logic gate operations (AND, OR, etc.). These operations could be performed in terms of different types of input characteristics, and here we choose three that are often relevant in neuroscience—the difference in magnitude, timing, and phase of two input signals. In particular, timing and phase add a new type of computing, as they are event-driven, require recurrent connections, and go beyond well-trodden digital logic. To give an explicit example, during one computation, the CRIREL circuit would be comparing one aspect of two input signals, and produce a response. For example, performing an AND operation in terms of input magnitude means that the circuit will only report "on" whenever two input values have the same magnitude, and will report "off" otherwise. We found it natural to classify possible unique types of outputs given these two inputs in terms of logical truth tables (i.e., AND, OR, XOR, etc.). It is important to stress that this goes beyond just reproducing digital logic; we will demonstrate that temporal and phase processing has the capability of detecting events. Moreover, computing can be done downstream of these events. Furthermore, classifying things in terms of a logical operation has the added advantage of cleanly classifying all the different ways the network operates upon its input. We show that for all three types of input classes, we are able to generate all 8 possible nontrivial logic truth tables. Finally, we show all 24 unique functions coexist for a given synaptic weight and that changes in functionality arise solely from the change in the bias current.

Results

The CRIREL circuits

To examine flexibility in small neural networks, we start with a CRIREL microcircuit [Fig. 1a][9]. This microcircuit is simultaneously near two cusp bifurcations [Fig. 1b][9], and as such, is capable of a simple two "bit" version of working memory [Fig. 1b, c]. Here, the"bits" are loosely whether one of the inhibitory neurons is ON or OFF. Note that two identical inputs pushed the memory circuit into a (1, 1) state, while asymmetric inputs pushed the memory circuit into a (1, 0) or (0, 1) state.

In order to compute neuromorphic inputs, we add two input neurons, which convert square impulses or sinusoidal waves into spike trains [Fig. 1e]. These neurons are solely for the purpose of making sure that the input into the circuit is biologically realistic. We also provide an output neuron, this is to concretely categorize the variable firing rate in the excitatory subsystem as a binary output. The output is considered one if it fires, and zero if it does not.

We compose the CRIREL circuit out of two different neuron implementation, the leaky integrate-and-fire neuron (LIF) and the Izhikevich neuron (IZH) with parameters chosen such that it is an integrator (See methods for parameters [Supplementary Tables 3, 5][42]). Given that the LIF neuron is an integrator[49], we set the IZH neuron as an integrator rather than a resonator to provide a more apples-to-apples comparison. Moreover, this emphasizes that the flexibility arises from the network itself, not the neurons' computational complexity.

Taking this two-bit classification as a starting point, we can classify functions in terms of binary logical operations. We take this input classification scheme, but associate the network with either an ON state or an OFF state as measured by the collective activity of the excitatory subsystem. This gives rise to eight unique nontrivial input combinations, AND, OR, XOR, NAND, NXOR, NAND, IMP, and NIMP [Fig. 1d and Supplementary Table 1], where the two inputs into the circuits can be classified as 1 or 0. The binary pair (1, 1) references if the inputs are "roughly" the same. Likewise, we can consider whether an out neuron is on or off as a binary output of 1 or 0. Moreover, we consider three unique types of inputs, the differences in magnitude between two square pulse inputs [Fig. 1Di], the differences in timing between two square pulse inputs[Fig. 1Dii], and the differences in the phase of two sinusoidal inputs [Fig. 1Diii].

Magnitude logic

The most straightforward type of input to consider is the differences in magnitude between two square pulse inputs [Fig. 2a]. After being fed into the input layer, the magnitude of the square pulses is converted into a difference in firing rate, and roughly corresponds to different magnitudes of post-synaptic excitatory input. We consider the input to be (1, 1) if the magnitudes A are roughly equal. Likewise, if the inputs are different by a fixed amount, ΔA they are considered (1, 0) or (0, 1). An absence of input is considered (0, 0). It is important to note that the magnitude A and the difference in magnitude ΔA are both free parameters [Supplementary Fig. 1].

As a concrete demonstration, we begin with an AND gate. For the particular example, we choose the magnitude for 1 to be $A = 1\ nA$ for the LIF neuron. We set $\Delta A = 0.3$ nA. When the network is presented with the input protocol, we see that the output neuron fires only when the signal is in the (1, 1) state [Fig. 2b]. Simply by changing the bias currents I_{bias} (see methods), we can change the gate into an XOR gate [Fig. 2c], where during the (1, 1) signal, the output is silent, but fires whenever the signal is in the (1, 0) or (0, 1) signal. Next, we show how we can change the bias currents, from AND to XOR in the middle of the input protocol [Fig. 2d] and the gate switches with a latency of 40 ms or about two spikes [Fig. 2e]. A majority of the transitions are possible, but a few require global reset (strong negative input)[Supplementary Fig. 2].

Next, we show that all eight unique logic gates exist for at least some set of bias currents. We test this for LIF neurons [Fig. 3a] and for Izhikevich (IZH) neurons in integrator mode [Fig. 3b]. The bias currents used to produce these eight gates can be found in the supplement [Supplementary Tables 7, 8]. We provide the parameter sweep for six of the logic gates as a two-dimensional sweep of excitatory subsystem bias and inhibitory

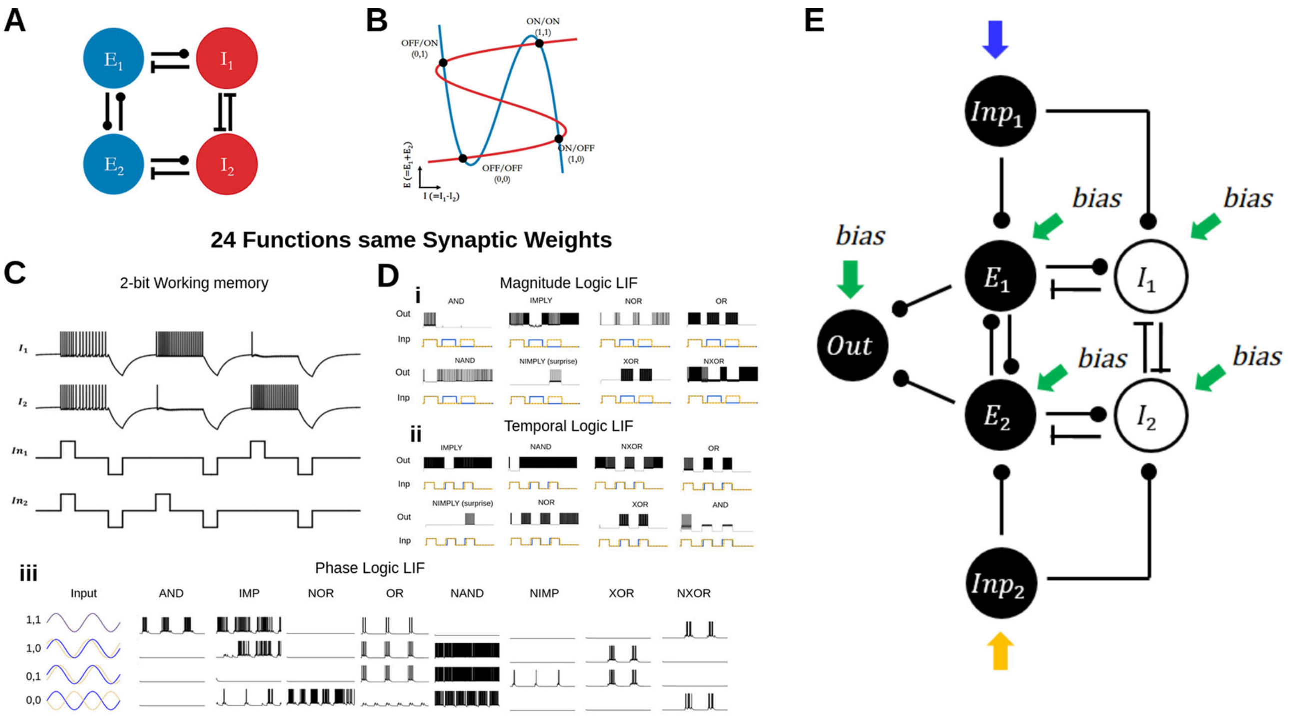


Fig. 1 | The CRIREL circuit. **A** The circuit consists of two mutually coupled excitatory neurons connected to two mutually coupled inhibitory neurons. **B** The 2D reduced phase plane of the firing rate version of the network. The blue nullcline is the average excitation, and the red nullcline is the difference between inhibitory neurons. The particular setup is capable of storing 2-bit working memory. **C** LIF implementation of the 2-bit memory storage. Depending on the input into the network, the network can either store ON/ON (1, 1), ON/OFF (1, 0), OFF/ON (0, 1), or OFF/OFF (0, 0). The network can store this information until a negative reset signal. **D** Twenty-four unique functions with the same synaptic weights. Only the input style and input strength change. All twenty-four distinct functions are categorized based on logic tables and input types. All twenty-four functions share the same synaptic weight; only input type and bias current change. **D**i The eight unique functions processing inputs with different magnitudes. **D**ii The eight unique functions processing inputs with different timing offsets. **D**iii The eight unique functions are processing two periodic inputs with different phase offsets. **E** The full circuit diagram to implement neuromorphic logic gates. Two input neurons are used to convert square pulses into spike trains. The output neuron is used to unambiguously map a two-dimensional input to a one-dimensional output. The input and output neurons are not essential for the computation of the CRIREL circuit.

subsystem bias [Fig. 3c, d]. Because IMP and NIMP require asymmetric bias currents, they don't fit neatly into a two-dimensional sweep, and are not shown.

It is important to note that while each gate has a region of parameter space, there are areas where the order of inputs matters. This is because hysteresis and bistability cause the preceding input to affect the computation in the next input. As such, we mark these as unclear (gray) in the parameter space [Fig. 3c, d].

Most of the logic gates are robust against variation in the bias currents. This robustness can be quantified as the square root of the area of each gate's parameter region in the two-dimensional sweep [Fig. 3c] of excitatory subsystem bias, and inhibitory subsystem bias [Fig. 3e]. This roughly quantifies the change in bias that each gate can tolerate. Some gates, like XOR and NAND are very robust, while NXOR and NOR, require greater precision in bias current.

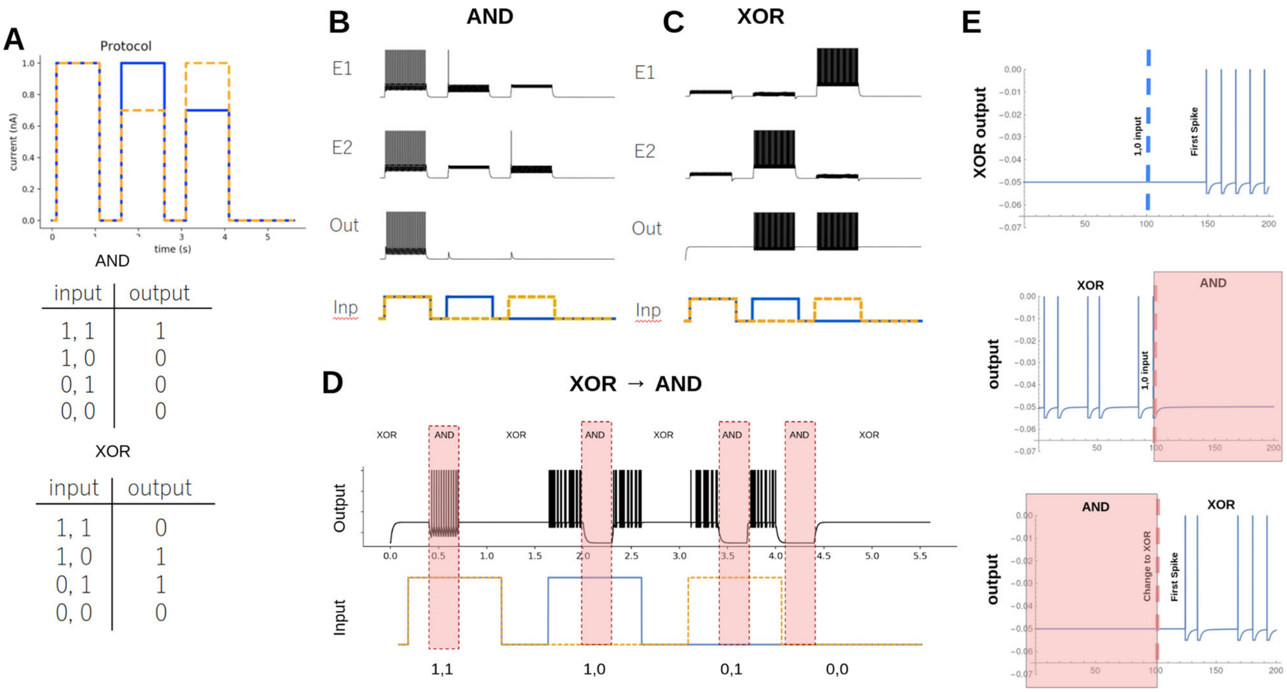
The gates also respond rapidly to the presentation of input. To quantify response time, we measure how long it takes for the "0" (OFF) response of the output neuron to switch to a "1" (ON) response. Here, we measure the time difference from when the input is presented to the network to the timing of the first spike in the output neuron. We sampled multiple different bias current configurations, some are faster than others, but all roughly respond in a few tens of milliseconds, which matches the scale of the membrane time constant of the neurons in the models.

Temporal logic

The CRIREL logic gates are not limited to processing the differences between the magnitude of two inputs, but can process the differences in the timing between two inputs [Fig. 4a]. Here we define two inputs arriving at the same time as (1, 1). Likewise, two inputs arriving at different times are (1, 0) and (0, 1). As before, the absence of any signal is considered (0, 0). Note that the magnitude of all inputs is the same.

To demonstrate, we show an NXOR gate (not exclusive or) [Fig. 4b]. Here, the network and, thus, output neuron fires in the absence of input (0, 0). Whenever the signal (1, 1) is present, the firing rate increases. However, whenever two signals arrive at different times (1, 0) and (0, 1), the circuit turns itself off, and the output neuron is silent. We can easily transform the network into any other of the eight logic gates. Here, we show the NAND gate [Fig. 4c], again, the absence of signal (0, 0) is firing, but the network fires when there is a timing difference (1, 0) and (0, 1) and is silent when the timing is the same (1, 1). We also highlight the NIMP gate, which is the logical negation of the Imply gate [Fig. 4d]. This gate shows no response if the timing is perfectly synchronized (1, 1) or the blue signal leads (1, 0), however shows a response if the yellow signal leads (0, 1). We refer to this also as a surprise gate, as one can imagine if the blue signal is trying to predict the yellow signal (with some time lag) the network remains silent. However, if the prediction fails, the network turns on to alert to a failure of prediction. Finally, we conclude with an XOR gate example [Fig. 4e]. It is worth stressing that these examples coexist with the same synaptic weights, only the bias currents change [Supplementary Table 9].

Again, we show that all eight unique logic gates exist for at least some set of bias currents, and test this for both LIF neurons [Fig. 5a] and for IZH neurons [Fig. 5b]. The bias currents used to produce these eight gates can be found in the supplement [Supplementary Tables 9, 10]. We provide the parameter sweep for 6 of the logic gates as a two-dimensional sweep of



AND

input	output
1, 1	1
1, 0	0
0, 1	0
0, 0	0

XOR

input	output
1, 1	0
1, 0	1
0, 1	1
0, 0	0

Fig. 2 | Magnitude logic. A Input protocol for the magnitude logic gate. The difference between the magnitude of the square pulses determines the (1, 1), (1, 0), (0, 1), and (0, 0). The truth tables for AND and XOR are shown. **B** The LIF version of an AND gate. The activity of the excitatory neurons (E1 & E2) and the output neuron (Out) are shown. A cartoon representation of the input protocol (Inp) is also shown. **C** Same as (**B**) but for an XOR gate. **D** An example of the CRIREL being converted back and forth between XOR and AND gates on the fly during the signal input. **E** The latency of switching output for the input a (0, 0) to a (1, 0) in an XOR gate is 47 *mS* or about five spikes. Latency of gate switching, which occurs on the scales of 20 ms or about two spikes.

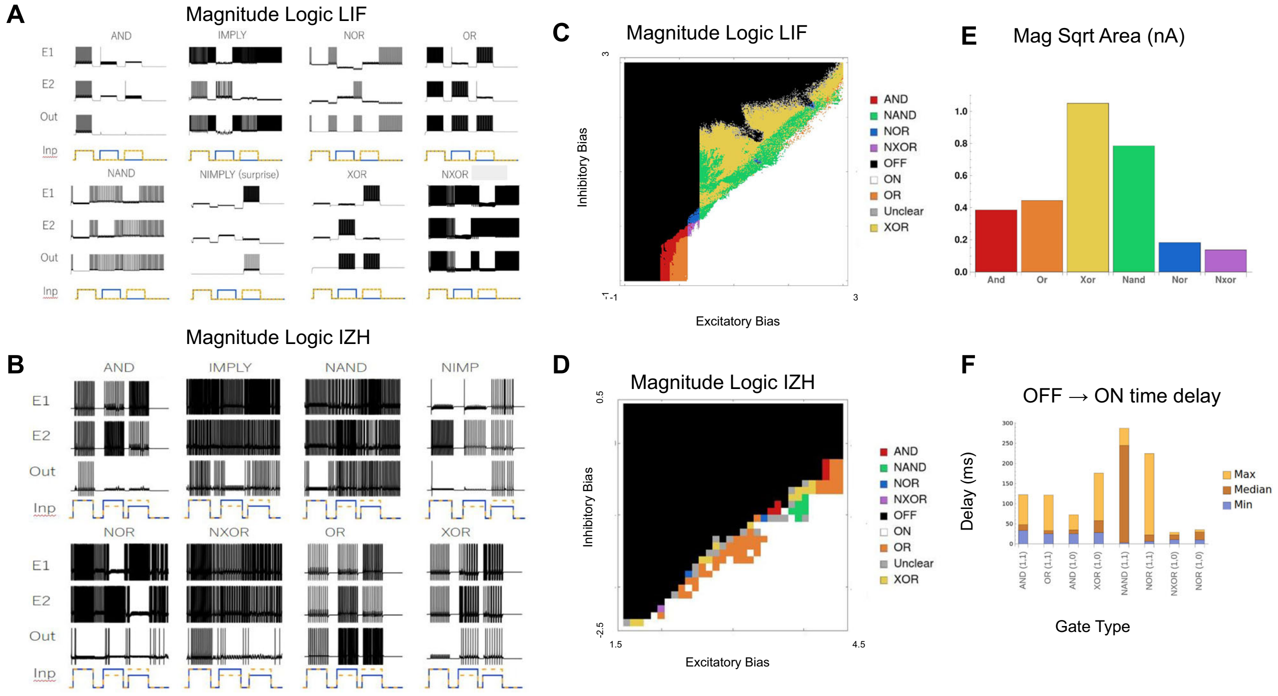
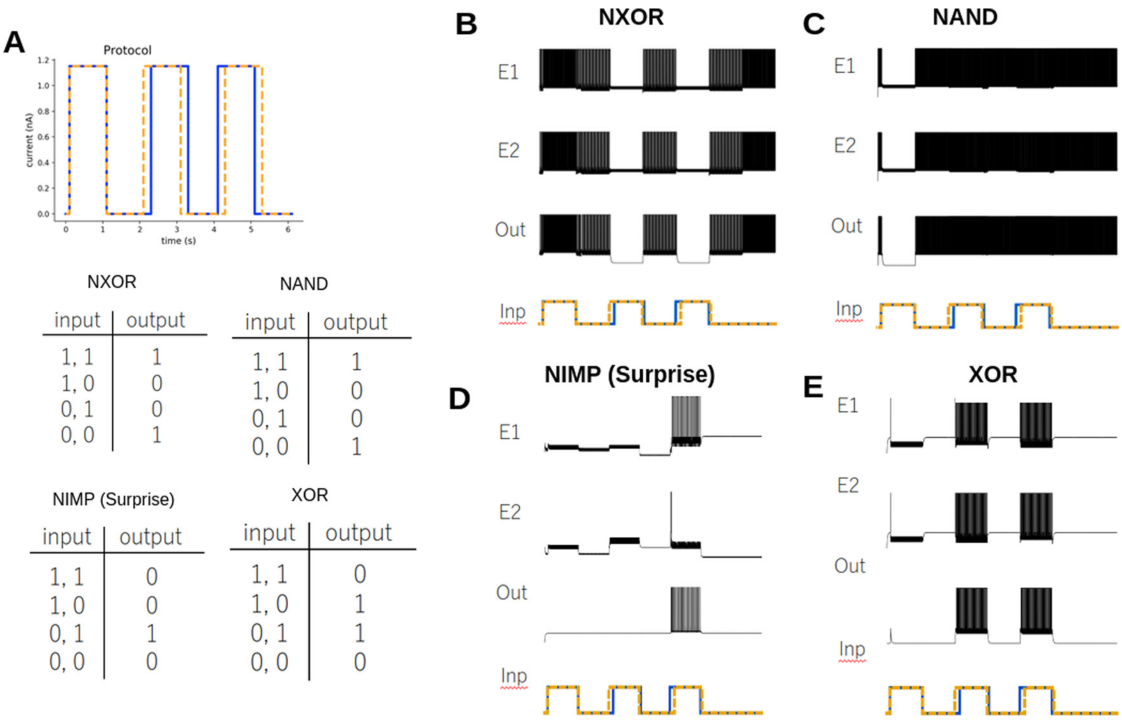


Fig. 3 | All magnitude gates. A All eight-magnitude logic gates for the LIF version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. **B** All eight-magnitude logic gates for the IZH version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. **C** A sweep through the inhibitory bias and excitatory bias for the LIF model. Each gate is labeled based on its color. **D** A sweep through the inhibitory bias and excitatory bias for the IZH model. Each gate is labeled based on its color. **E** The robustness of each LIF gate in terms of bias current. It is calculated as the square root of the area for each gate in (**C**). Larger values represent more robust gates. **F** The average time from a silent OFF state for the output neuron to the first spike of an ON state for the output neuron. We sampled different biases from the sweep in (**C**).



NXOR

input	output
1, 1	1
1, 0	0
0, 1	0
0, 0	1

NAND

input	output
1, 1	1
1, 0	0
0, 1	0
0, 0	1

NIMP (Surprise)

input	output
1, 1	0
1, 0	0
0, 1	1
0, 0	0

XOR

input	output
1, 1	0
1, 0	1
0, 1	1
0, 0	0

Fig. 4 | Temporal logic. A Protocol for the temporal logic gate. The difference between the timing of the square pulses determines the symbols (1, 1), (1, 0), (0, 1), and (0, 0). The difference in timing is 10 ms. The difference between signals is exaggerated for clarity. The Truth tables for NXOR, NIMP, NAND and XOR are shown. **B** The LIF version of an NXOR gate. The activity of the excitatory neurons (E1 & E2) and the output neuron (Out) are shown. A cartoon representation of the input protocol is shown. **C** The LIF version of a NAND gate. The activity of the excitatory neurons and the output neuron are shown. **D** The LIF version of the NIMP gate. This gate is the logical negation of the Imply gate. **E** The LIF version of the XOR gate.

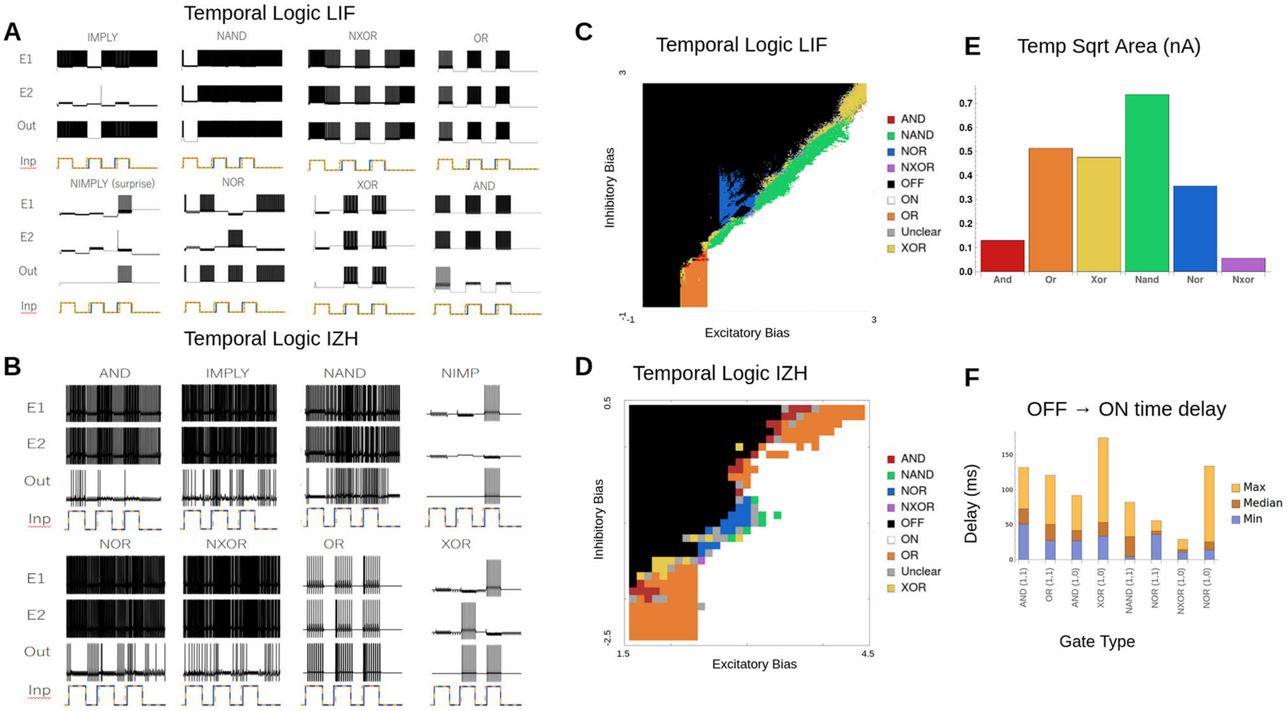
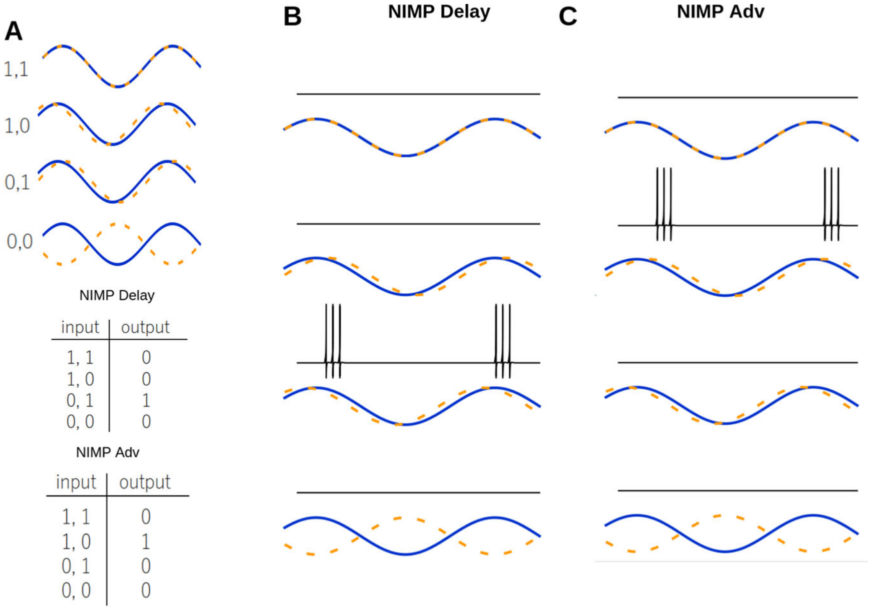


Fig. 5 | All temporal gates. A All eight temporal logic gates for the LIF version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. **B** All eight temporal logic gates for the IZH version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. **C** A sweep through the inhibitory bias and excitatory bias for the LIF model. Each gate is labeled based on its color. **D** A sweep through the inhibitory bias and excitatory bias for the IZH model. Each gate is labeled based on its color. **E** The robustness of each gate in terms of bias current. It is calculated as the square root of the area for each LIF gate in (**C**). Larger values represent more robust gates. **F** The average time from a silent OFF state for the output neuron to the first spike of an ON state for the output neuron. We sampled different biases from the sweep in (**C**).



NIMP Delay

input	output
1, 1	0
1, 0	0
0, 1	1
0, 0	0

NIMP Adv

input	output
1, 1	0
1, 0	1
0, 1	0
0, 0	0

Fig. 6 | Phase logic. A Protocol for the phase logic gate. The difference between the phase of the sine waves determines the input state, i.e., (1, 1) synchrony, (1, 0) phased advanced, (0, 1) phase delayed, and (0, 0) anti-synchronized. The truth tables for NIMP delayed, and NIMP advanced are shown. **B** The IZH version of a NIMP delayed gate. The activity of the output neuron is shown. A cartoon representation of the input protocol is shown. **C** The IZH version of a NIMP advanced gate. The activity of the excitatory neurons and the output neuron are shown.

excitatory subsystem bias, and inhibitory subsystem bias [Fig. 5c, d]. As before, IMP and NIMP require asymmetric bias currents, they don't fit neatly into a two-dimensional sweep.

We can test robustness by considering magnitude *A* and timing difference Δ*t* as free parameters (Supplementary Fig. 3). It is important to note that the synaptic weights are the same as the synaptic weights in the magnitude example, only the input protocol was changed.

Phase logic

Given the ability to look at the relative timing of two inputs, it is natural to ask if computation can be done on phase differences between two periodically repeating inputs. The input neurons take a pure sinusoidal signal [Fig. 6a]. Here it is important to note we do not convert the sine-wave into a spiking signal, rather we feed the raw sine-wave into the CRIREL network directly.

As expected, the definition of the inputs and output change. We consider (1, 1) to be when the two input waves are synchronized. Similarly, we consider (0, 1) to be phased delayed and (1, 0) to be phased advanced. Interestingly, we can define (0, 0) as two anti-synchronized inputs. Here, an output neuron is considered ON if, at any time during a full cycle of the sinusoidal input, the output neuron spikes. Often, the output neuron will fire during the maximum of the input wave [Fig. 6b, c], but for anti-synchronized (0, 0), the output neuron fires during the average value [Fig. 7a NOR].

As a concrete example, we point out three examples: AND [Fig. 7a AND], NIMP advanced [Fig. 6b] and NIMP delayed [Fig. 6c], and NOR [Fig. 7a NOR]. For example, AND acts as a coincidence detector and fires only when the two signals are synchronized [Fig 7a AND]. NIMP (or Not Imply), we call the surprise detector [Fig. 6b], as if you imagine one wave predicting the other wave, then the output neuron will only fire when the prediction lags behind the measurement, that is, the prediction is no longer accurate. More concretely, that means NIMP's output neuron fires whenever it is in the (1, 0) state, not the (0, 1). Notice that the distinction between (1, 0) and (0, 1) is arbitrary, and as such, there are two symmetric versions of NIMP that can be easily switched between by flipping the asymmetry present in the bias currents [Fig. 6c]. The NOR gate [Fig. 7a] only fires when the two waves are perfectly anti-synchronized. It is silent in the other three cases.

We also show that all eight unique logic gates exist for phase-type computations, and test this for both LIF neurons [Fig. 7a] and IZH neurons [Fig. 7c]. The bias currents used to produce these eight gates can be found in the supplement (Supplementary Tables 11, 12). We provide the parameter sweep for six of the logic gates as a two-dimensional sweep of excitatory subsystem bias, and inhibitory subsystem bias for LIF [Fig. 7b, d]. Because IMP and NIMP require asymmetric bias currents, they don't fit neatly into a two-dimensional sweep.

It is also important to discuss how precise the synaptic weights must be. The gates would be useless if they require a specific synaptic weight and cannot tolerate any deviation from this value. We find that most gates are robust up to 10% difference in the synaptic weights [Supplementary Fig. 4].

Application: adder circuit

Flexible gate switching is a unique property of CRIREL logic gates. To demonstrate its advantages, we recreated the well-documented 4-bit ripple carry adder circuit, which is a digital logic circuit used for performing multi-bit binary addition. It is composed of four 1-bit full adders [Fig. 8a]. A full adder takes in three inputs, the two summands, and the carry from the previous digit [Fig. 8b]. This is fed into two CRIRELs, one for computing the sum and the other for computing the carry. We add a buffer neuron to regularize the output firing rate. Note that a traditional transistor-based adder requires nine NAND logic gates [Fig. 8c][55], while only two neuromorphic logic gates are required.

Here, it is important to note that the two summands are the traditional magnitude inputs from above. Here, the carry induces a change in the bias current of the CRIRELs to switch the logic gate from AND to OR for the carry CRIREL, and XOR to NXOR for the sum CRIREL [Fig. 8d]. The bias currents used to switch can be found in [Supplementary Table 13] and are graphically represented as well [Fig. 8e].

It is also worth noting that to ensure output amplitude matches output magnitude, we added a buffer neuron to match the output firing rate (i.e., magnitude) to the input firing rate for AND, OR, XOR, and NXOR logic to work. It works by having a sharp threshold such that it reaches the maximum firing rate with any input from the output neuron. To see the parameter of the buffer neuron, see the methods [Supplementary Table 2]. One particularly interesting implication of the full-adder is that CRIREL circuits can be combined together to create cascading logical operations. This is by no means limited to magnitude logic (as with the full adder example), but extends to phase and temporal computation as well [Supplementary Fig. 5].

To implement a 4-bit ripple adder, we connect four 1-bit full adders together [Fig. 8f]. A 1-bit adder requires at least 50 ms to generate a correct response time [Fig. 8g]. Carry propagation delay necessitates a minimum 200-ms delay for a 4-bit ripple carry adder.

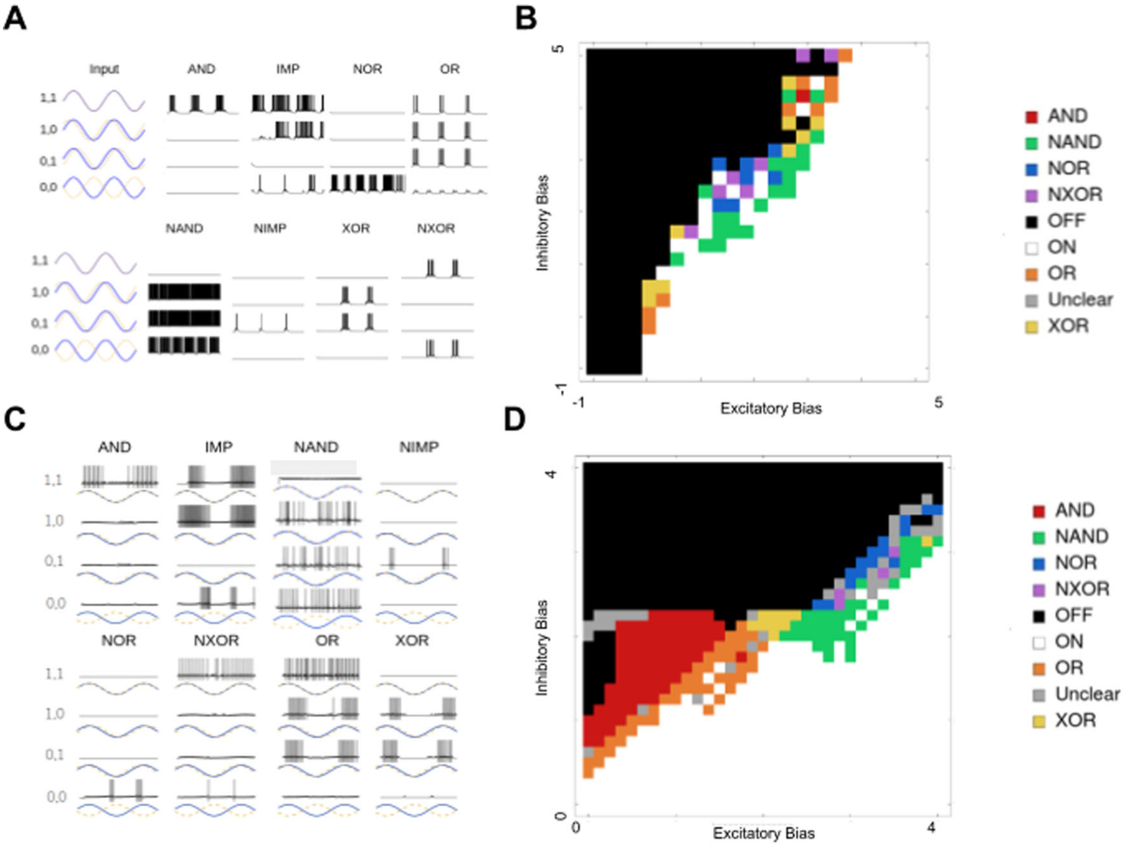


Fig. 7 | All phase gates. A All eight phase logic gates for the LIF version of the CRIREL. Both IMP and NIMP have a mirror image that is not shown. **B** A sweep through the inhibitory bias and excitatory bias for the LIF model. Each gate is labeled based on its color. **C** All eight phase logic gates for the IZH version of the CRIREL. **D** A sweep through the inhibitory bias and excitatory bias for the IZH model. Each gate is labeled based on its color.

We investigated CRIREL's noise tolerance, which is essential to a network's usability. We applied Gaussian noise to the bias and measured the accuracy of the logic operation (see Materials and Methods). We quantified both absolute noise (measured in nA) and relative noise (percentage of bias current). Results show CRIREL-based adders have a 0.1 nA tolerance for absolute noise and a 10% tolerance for relative noise [Fig. 8h] (NXOR is an exception, ranging between −10 to +5%).

Application: reconfigurable motion detection

Conventional motion detectors such as the Hassenstein-Reichardt model only detect a fixed direction it is set to detect[56]. Our NIMP gates can perform conditional motion detection by ignoring the panning motion and detecting only the moving object in the scene. We demonstrate how NIMP gates can be reconfigured to detect objects moving against optical flow induced by a panning camera. This functionality relies on two NIMP gates that are antisymmetric counterparts of each other [Fig. 9a]. NIMP gates can be used to detect either rightward or leftward optical flow. For simplicity, we can consider optical flow on a one-dimensional pixel array [Fig. 9b]. Here, we have a collection of white pixels moving either rightward or leftward. The circuit compares the pixel value of two adjacent pixels. Depending on the chosen bias currents, the circuit is either rightward-selective or leftward-selective, firing spikes whenever bright objects pass by the two pixels being compared. It achieves selectivity by comparing the phase delay between the left and right pixels [Fig. 9b]. Depending on the bias of the network, the network responds only if the right pixel lags the left pixel, or if the left pixel lags the right pixel. We can now consider a panning camera, again considering a 1D pixel array. The camera either pans right or left, inducing a flow in the opposite direction. At 1000 ms, an object appears moving against the optical flow induced by the panning [Fig. 9c]. The NIMP gate's directional sensitivity can be tuned to pick up either the object moving against the flow [Fig. 9d left] or the flow itself [Fig. 9d right]. They can be reprogrammed depending on the direction of the panning camera. To detect an object moving with the flow, the directional selectivity is set to be opposite the flow induced by the panning camera. While this is just a simple setup to demonstrate a proof-of-concept, the key observation is that the camera pan can send a signal to the CRIREL circuits and reconfigure the network's receptive field to be leftward or rightward selective. There is no need to change synaptic weights to change the network's receptive field.

Independence of neuronal unit

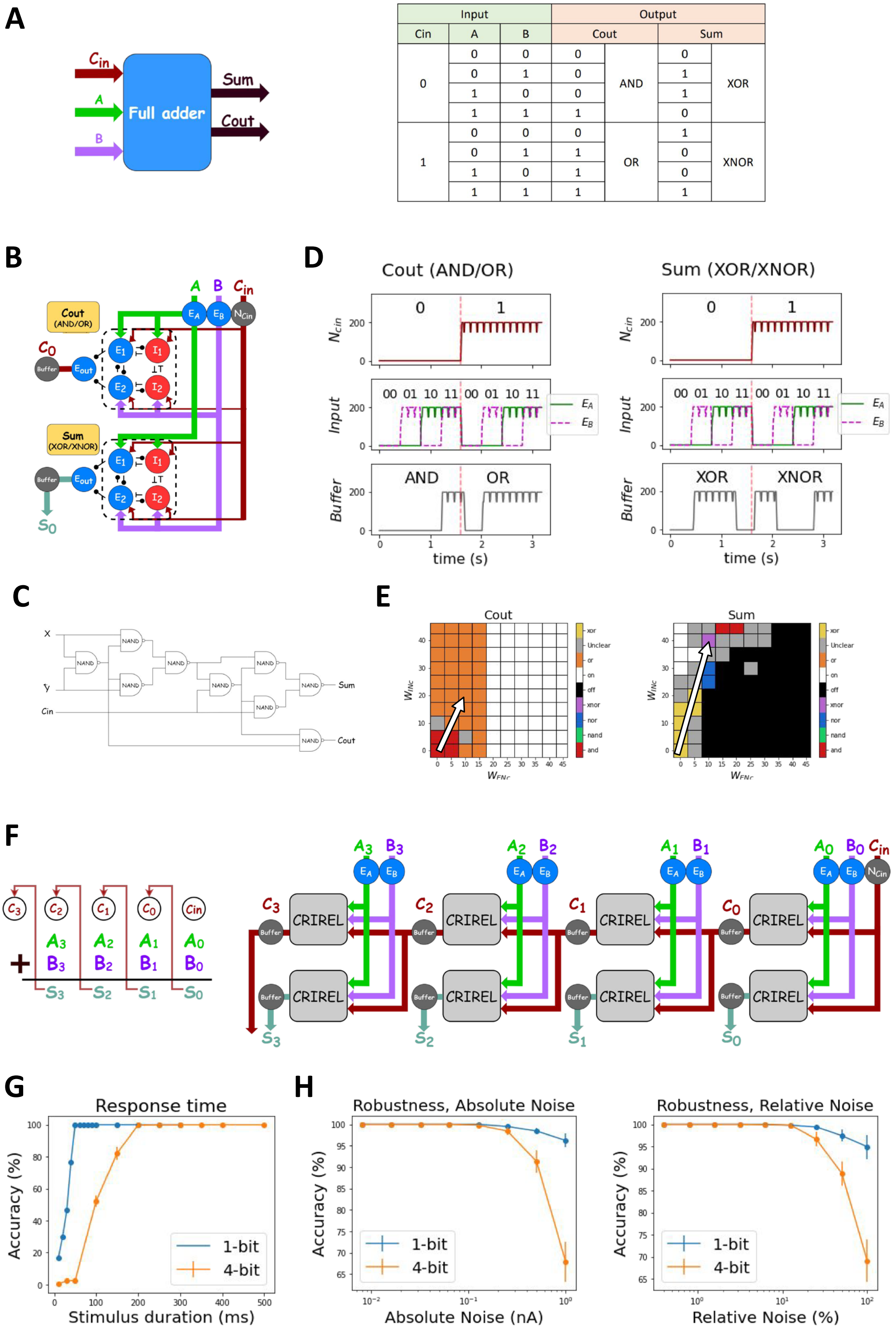
We have stressed that the underlying neuron model that the CRIREL circuit is composed of does not matter. Any neuron with a monotonic saturating IF curve can form a flexible neural network. However, the implementation need not be spiking. Components with saturating non-linearity can also perform flexible logic. Any system of the following form can have flexible and reconfiguable logic.

$$\tau\frac{de_1}{dt} = -e_1 + f(g_{ee}e_2 - g_{ie}i_1 + b_e) \tag{1}$$

$$\tau\frac{de_2}{dt} = -e_2 + f(g_{ee}e_1 - g_{ie}i_2 + b_e) \tag{2}$$

$$\tau\frac{di_1}{dt} = -i_1 + f(g_{ei}e_1 - g_{ii}i_2 + b_i) \tag{3}$$

$$\tau\frac{di_2}{dt} = -i_2 + f(g_{ei}e_2 - g_{ii}i_1 + b_i) \tag{4}$$



Input			Output			
Cin	A	B	Cout		Sum	
0	0	0	0	AND	0	XOR
	0	1	0		1	
	1	0	0		1	
	1	1	1		0	
1	0	0	0	OR	1	XNOR
	0	1	1		0	
	1	0	1		0	
	1	1	1		1	

Fig. 8 | An adder application for CRIRELs. **A** The schematic and the truth table of full adder. Carry-out (Cout) toggles between AND and OR gates, while the summation (Sum) toggles between XOR and NXOR gates. **B** The networks of 1-bit full adder. **C** A classical 9-gate 1-bit adder, note that it has seven more required logic gates than the gate switching adder. **D** The output firing rates of two CRIRELs. **E** Parameter space for the control neurons, when the neuron is off, the network switches from the starting gate AND and XOR to NXOR and OR. The synaptic weight of the control neuron is tuned such that the resulting constant current changes the bias current to switch to the correct gate. **F** The schematic and network of 4-bit ripple carry adder. **G** The response time of the adder. A 1-bit adder requires a response time of 50 ms, and a 4-bit ripple carry adder needs 200 ms. **H** The adder composed of CRIRELs can tolerate an absolute noise of 0.1 nA and relative noise of 10% in its bias.

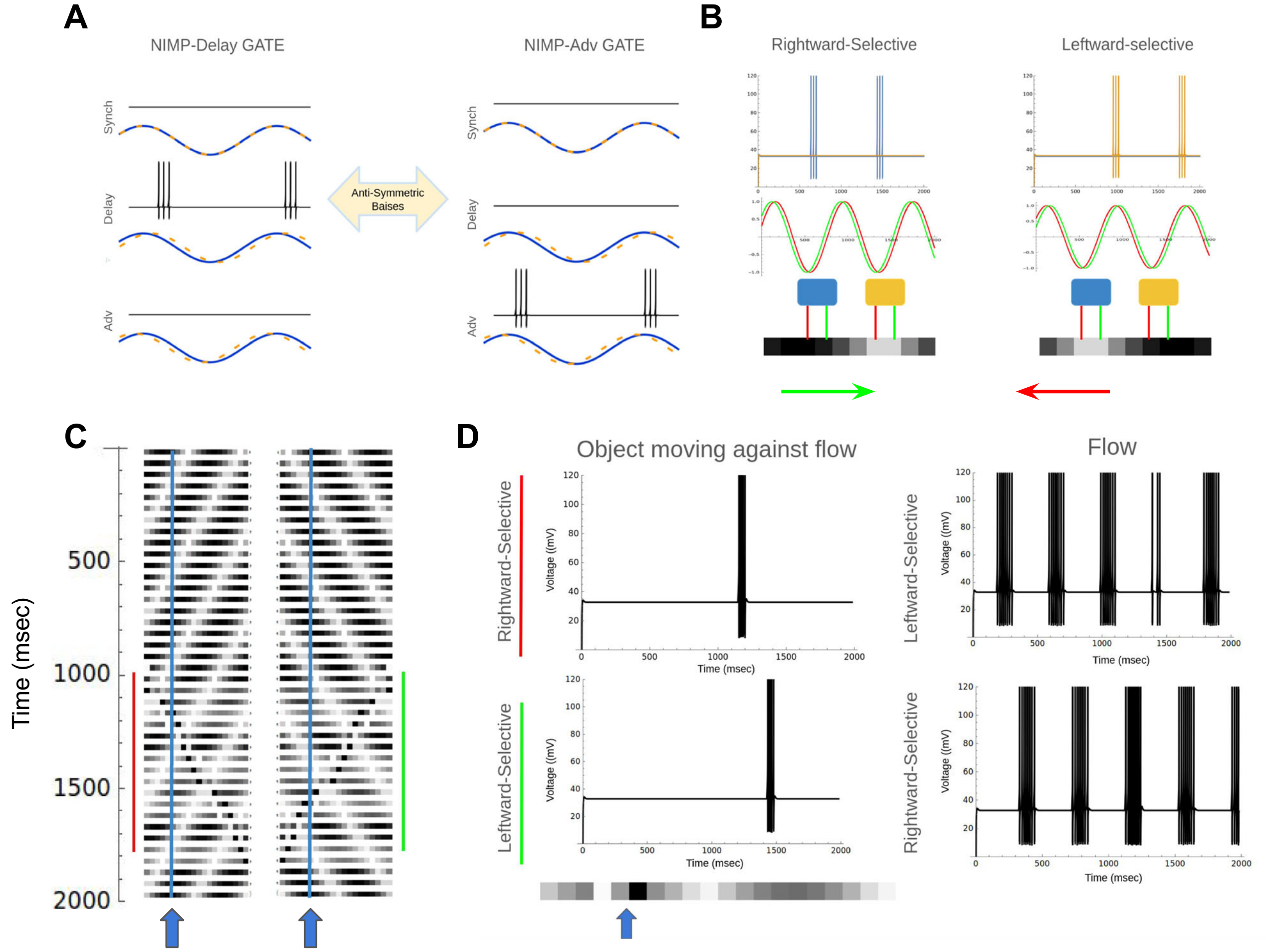


Fig. 9 | Detecting motion against a moving background. **A** The not imply (NIMP) gate has two antisymmetric versions that can be easily switched between by flipping the bias current asymmetries. **B** Two circuits with different antisymmetric bias currents are sensitive to either rightward-moving white pixels or leftward-moving white pixels. **C** Two examples of a leftward/rightward moving flow with an object moving rightward/leftward at 1000 ms. **D** Circuits with NIMP biases comparing two adjacent pixels (indicated by the blue and yellow arrows). The blue circuit detects rightward motion, the yellow circuit detects leftward motion. The network can distinguish between flow and the object moving against the flow.

$$\tau \frac{do_1}{dt} = -o_1 + f(g_o(e_1 + e_2)) + b_o) \tag{5}$$

Here we show two implementations of f given by a clipped ReLU [Fig. 10a] and a sigmoid function [Fig. 10b].

$$f(x) = c\mathrm{ReLU}(x) = f(x) = \begin{cases} 0 & \text{if } x<0 \\ x & \text{if } 0 \geq x \geq c \\ c & \text{if } x \geq c \end{cases} \tag{6}$$

$$f(x) = \mathrm{sig}(x) = \frac{1}{1 + e^{-\frac{x}{a}}} \tag{7}$$

As before, we can test the three different input types: magnitude [Fig. 10ci], temporal [Fig. 10cii], and phase [Fig. 10ciii] for both the sigmoidal function and the clipped ReLU. Again, all six symmetric logic gates can be found by sweeping through b_e and b_i. It is important to note that there can be oscillatory solutions present. While outside the scope of the paper here, these are useful for pace-making and central pattern generation standpoint (see ref. 9) for a thorough investigation of oscillatory behavior.

Discussion

In this study, we conclusively show that small spiking recurrent neural microcircuits are incredibly flexible. Here, by picking a set of synaptic weights such that the bias currents can uncontrollably unfold the double-cusp bifurcation, we are able to switch between an incredibly diverse range of different operations. To concretely define unique functions, we used logical truth tables to associate each function with a logic gate. Moreover, we showed that these logic gates can process three different types of inputs, namely, magnitude, temporal, and phase. Furthermore, we found that the gates are robust against membrane noise, and operated over a range of synaptic weights. Strikingly, we showed that these results were independent of the neuron implementation, as we found working versions in LIF, IZH neurons (in the integrator regime), and firing rate models. While we tested the network with all integrator neurons[49], it would indeed be interesting to test the network with the IZH neurons set as resonators[42].

We also investigated two new types of information processing: differences in the arrival of signals, temporal, and differences in the phase of periodic signals. We showed that the flexible gate switching can be used to create a 4-bit ripple carry adder. Additionally, we showed how to define logic with the differences between the signals. Moreover, we showed that this logic can be used to detect events. Specifically, these networks can rapidly and flexibly reconfigure to detect either a rightward traveling white pixels, or a

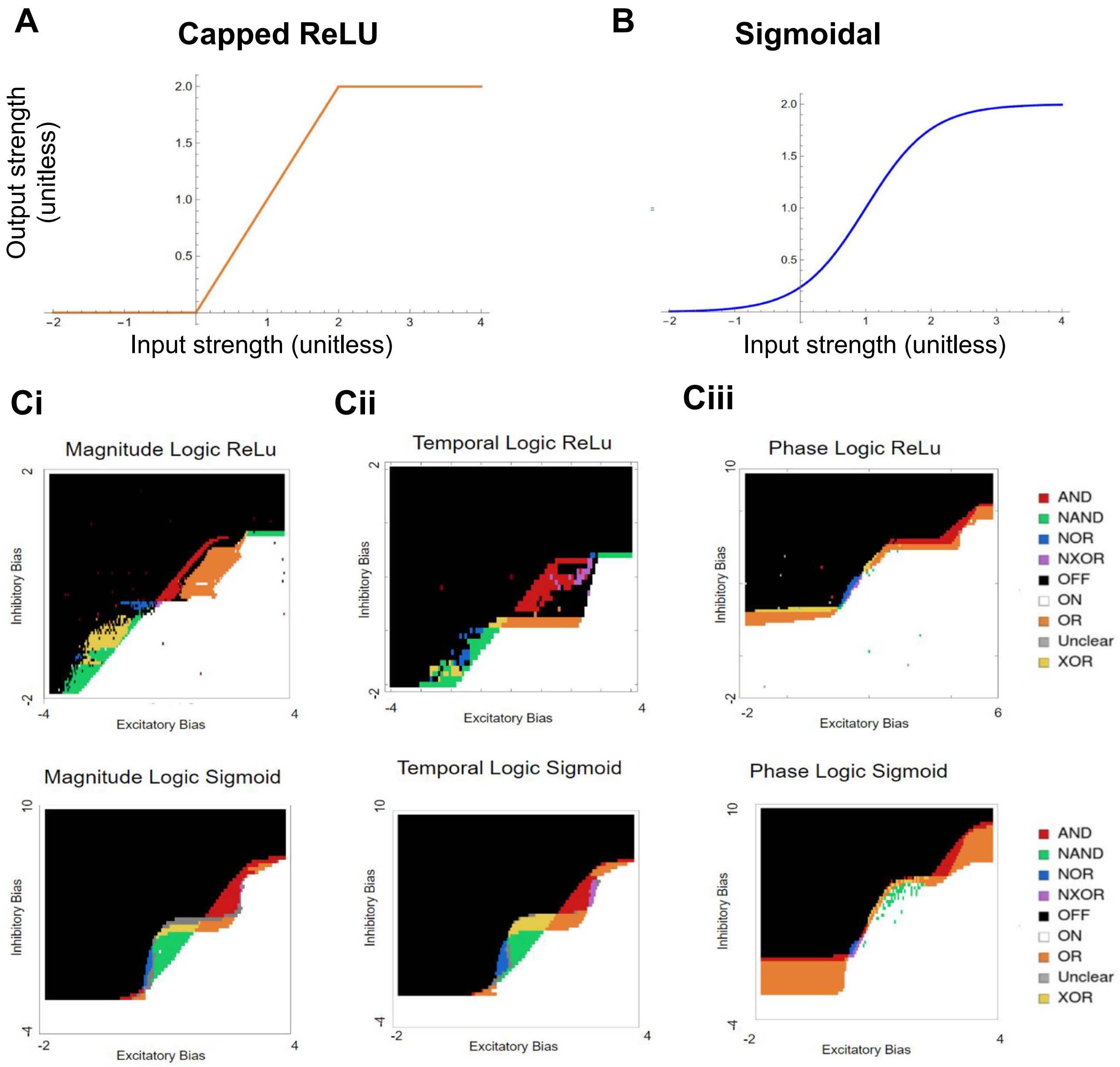


Fig. 10 | Examples of saturating rate fire neurons in CRIREL circuit. **A** Capped ReLU function activation curve. **B** Sigmoidal function activation curve. **C**i A sweep through the inhibitory bias and excitatory bias for magnitude computation. Each gate is labeled based on its color. **C**ii A sweep through the inhibitory bias and excitatory bias for temporal computation. **C**iii A sweep through the inhibitory bias and excitatory bias for phase computation.

leftward traveling white pixels. Finally, we show that the reconfigurability is a property of the network, not the underlying neuronal model.

This is in line with previous research showing that recurrent neural networks are more flexible than their feedforward counterparts[9]. Specifically, this study takes advantage of a nearby cusp bifurcation[53]. This tracks with the plethora of studies that show networks near bifurcations are inherently more flexible[33,49–53]. Studies have been able to embed complex functions into networks that are near multiple bifurcations[33,38,48], and other studies have shown, that networks far away from bifurcations means that the network has redundant neurons[53]. Here, we push this idea to its limit by fitting 24 unique functions into a 4-neuron microcircuit. Additionally, we show that one can rapidly switch between these functions without changing the synaptic weights, making use only of the bias currents.

This provides a particularly powerful technique, as often switching functions is seen as a neuromodulatory process[10,19–21], or a synaptic weight process[36–40]. While clearly, these process can and do change a network's function, they are not the only means. Contextual information and top-down control have been shown to optimize[23,25,27,30,50,51,57], or even switch, the function of decision networks in the prefrontal cortex[13,58,59]. Here, we show that bias current (which can be thought of as weak constant input) can rapidly switch the function of a microcircuit, allowing a potential mechanism for integrating contextual information or top-down control. This is a tantalizing direction for future investigation, especially if a learning algorithm can be used to embed multiple tasks into recurrent networks.

Currently, feedforward spiking neural networks (SNNs) are also capable of computing logic operations[34,35], with modern implementations operating on single spikes. While these models excel in quick computation due to their single-spike operation, they are more susceptible to noise, as any variation in spike timing can disrupt the logic computation [Supplementary Table 14].

In these models, synaptic weights are precisely tuned for each logic gate, which helps minimize the neuron count and synapse number [Supplementary Table 14][34]. Despite this, no feedforward network currently can robustly and flexibly adjust to all possible gates by merely changing the bias current. More critically, no feedforward network can compute temporal logic.

This limitation arises because a feedforward network possesses one fixed point and lacks both working memory and temporal dynamics. After the input delay has passed, an input of (1,0) or (0,1) becomes indistinguishable from an input of (1,1). Consequently, distinguishing gates XOR and NXOR from always-off (OFF) and always-on (ON) gates becomes impossible.

While our network here has a particular neuromorphic implementation, we are not limited by these implementations. Our results show that any sigmoidal saturation is capable of generating a flexible and reconfigurable circuit. This leaves the door open to many potential applications. The sigmoidal saturation means reconfigurable electronic circuits could be made with memristors[4,5,46] or homojunctions[45]. Moreover, our circuit's compatibility with neuromorphic architectures could find use in neuromorphic chips[60–62], especially with liquid state machines[33,62]. Finally, given that the network was reconstructable with Izhikevich neurons[49] it is conceivable that our neural circuit can be grown in a real brain organoid[7,8].

Another potential avenue for future work involves the robustness of some of the gates. Gates like AND were the least robust of the operations, being the least tolerable to noise. However, it is known that gap junctions are capable of averaging out the activity levels of two neurons. If a gap junction (which is commonly observed in parvalbumin-containing neurons[63,64]) is added between the inhibitory neurons, this should expand the basin attraction associated with the (1, 1) state. A larger (1, 1) state should translate into more robustness to noise for the AND state, at the cost of robustness of the XOR state. This remains to be thoroughly tested.

While we tried to be as comprehensive with input processing types, we cannot be exhaustive. Several interesting directions remain for future work. Is there computation based on differences in frequency? Would a network composed of resonators be ideal for detecting differences in frequency? On the other hand, could the order in which symbols are presented (1, 1) then (0, 1) be different than (0, 1) then (1, 1)? Could one leverage this to construct finite-state machines? These are tantalizing open questions.

Finally, while we give a useful account of small microcircuits, the approach needs to be expanded to larger networks. Some studies have successfully embedded arbitrary motor commands into E/I balanced networks near multiple bifurcations[33]. It remains to be seen how many unique functions can be embedded in a larger network, and if this technique of unfolding the bifurcation in predictable ways can be of use. Using our full adder example, we demonstrated how CRIRELs can be interconnected to form cascading logic circuits, where the output of one circuit serves as the input for the next. We showed logic can be cascaded for magnitude (Fig. 8), phase, and temporal computation [Supplementary Fig. 5]. However, this is far from a comprehensive list of possible logic circuits. Future work is needed to explore other higher-order circuits composed of flexible CRIRELs. Applications can include motion control[1], motor planning[33], and event-based image processing[65,66]. Specifically, can learning be used to configure a group of 4-neuron CRIREL circuits, or even a larger, more general network? If so, that could open up new opportunities to minimize neuron count but maximize task number. This could have major consequences for edge-deployed neuromorphic computing.

To conclude, we made use of CRIREL networks near a double-cusp bifurcation, and were able to control the unfolding of the bifurcation with the bias currents. We demonstrated that these microcircuits have a rich repertoire of 24 unique functions. Ultimately, this provides a computational foundation for how neural adaptability can occur on timescales much shorter than plasticity. This is a novel way to control a neural circuit, and could lead to new types of computing, especially in the growing field of neuromorphic computing.

Methods

We make use of two neuron models, the Leaky Integrate-and-Fire model, and the Izhikevich model. This is to show that flexibility is a property of the network, not the underlying neurons. Moreover, it shows the flexibility is robust to model choice.

Leaky integrate-and-fire model and synaptic model

We implement our leaky integrate-and-fire (LIF) model using our in-house simulator Flysim[67]. Recall LIF neuron dynamics are given by the equation

$$C_m \frac{dV_i}{dt} = -g_L\left(V_i - E_L\right) + I_{\text{bias}} + \sum_j I_{i,j} \tag{8}$$

$$I_{ij} = \sum_j g_{ij} s_{ij}\left(V_i - E_{ij}\right). \tag{9}$$

V_i is the membrane potential of the ith neuron, C_m is the membrane capacitance, g_L denotes the leak conductance, and E_L represents the leak potential. Here, g_{ij} corresponds to the synaptic conductance between neurons. We use s_{ij} to denote the synaptic variable, and E_{ij} denotes the synaptic reversal potential. Most importantly, I_{bias} is the constant bias current we manipulate to generate the diverse set of logical operations.

We utilize an excitatory synapse (AMPA-like), referred to here as Exc, and an inhibitory synapse (GABA-like), referred to here as Inh, as the neurotransmitters for excitatory and inhibitory neurons. We refer to the two excitatory neurons as E1 and E2, and the two inhibitory neurons as I1 and I2 (see Fig. 1A). They are almost always considered symmetric, except when different inputs or bias currents are provided. The equations for the synapses are shown below, and parameters for various neurons are given in Supplementary Table 2.

$$\frac{ds}{dt} = -\frac{s}{\tau} + \sum_k \delta\left(t - t_k\right) \tag{10}$$

When the membrane potential exceeds the threshold voltage ($V_{\text{threshold}}$), a neuron generates a spike and then is reset to (V_{reset}). There is no change in voltage until after the refractory period lapses.

$$\text{when, } V_i > V_{\text{threshold}} \text{ then } V_i \overset{\text{spike}}{\rightarrow} V_{\text{reset}} \tag{11}$$

The parameters for the LIF neuron are shown in Supplementary Table 3. We must stress that the synaptic weights for every function are the same. The synaptic weights used for the LIF model are listed in Supplementary Table 4.

Izhikevich model

The Izhikevich neuron is a nonlinear generalized quadratic integrate-and-fire neuron[42,49]. We implement the Izhikevich neuron (IZH) in Mathematica. We use the following version of the IZH neuron

$$\frac{dV}{dt} = \frac{1}{k}V^2 - 3V + 40 + 100b - W + I_{\text{bias}} + \sum g_i S_i \tag{12}$$

$$\frac{dW}{dt} = a(bV - W) \tag{13}$$

$$\text{When } V > 130,\ V \rightarrow c,\ W \rightarrow W + d \tag{14}$$

Voltage was set to be akin to millivolts with an always positive voltage, as to aid eventual implementation on a neuromorphic chip. When the voltage hits 130, the system resets to voltage *c*, and the gating variable *W* is reset to $W + d$. Again, we use I_{bias} to denote the constant bias current we manipulate to generate the diverse set of logical operations. The synapse gating variables *S* were governed by the following equation

$$\tau \frac{dS}{dt} = S + \delta(t - t_{\text{spike}}) \tag{15}$$

where τ is a time constant, and δ is the Dirac delta function, raising the value of the synaptic gating variable *S* by one unit every spike. The parameters (which should be considered unitless) are given in Supplementary Table 5. We choose the parameters such that the neuron is an integrator.

Synaptic Weights between each neuron are given in Supplementary Table 6.

Protocol for binary inputs

To properly define a unique function, we use logic gates as a useful classification of different functions. Classically, the input of a logic gate operates on a binary input of two digits, either 0 or 1. There are four possible combinations of input (1, 1), (0, 1), (1, 0), and (0, 0) and two possible outputs, 1 or 0. There are 16 possible combinations of inputs and outputs [Supplementary Table 1].

However, a neural circuit is not constrained to be binary. Neurons can respond to many different properties of an input. Here, we define an (1, 1) input as two inputs that are approximately the same (see supplemental material for the precise range of sameness). If one input is larger in magnitude, it is (1, 0) or (0, 1). If one input arrives before the other (so-called temporal logic), we can also define it as (1, 0) or (0, 1). Finally, if one input is phase delayed to the other input, we can classify it as (1, 0) or (0, 1). For both magnitude and temporal logic, (0, 0) is defined as an absence of input. However, for phase logic, (0, 0) is defined as perfect asynchrony. The truth tables can be found in Supplementary Table 1.

In the LIF implementation, for both magnitude logic and temporal logic, we present the input neurons of the network with two square inputs. For the 1 symbol, the square wave has an amplitude of 5 nA. The 0 symbol has an amplitude of 3.5 nA (except during robustness testing).

In the IZH implementation, for both magnitude logic and temporal logic, we directly present the CRIREL circuit with the square pulse. For the 1 symbol, the square wave has an amplitude of 2.5. The 0 symbol has an amplitude of 2.5 (except during robustness testing). Recall that the Izhikevich neuron is unitless.

For phase logic, in both versions, we present a sinusoidal input directly to the CRIREL circuit. In the LIF model, the wave has a period of 40 ms, and an amplitude of 2 nA. Phase delay was defined as $\frac{\pi}{10}$ radians. For the IZH model, the wave has a period of 400 ms, and an amplitude of 1. Phase delay was defined as $\frac{\pi}{8}$ radians.

Categorizing logic gates

To categorize the logic gates for the two neuromorphic implementations, we used a protocol in which the input given to the circuit follows the following order: (0, 1), (1, 1), (1, 0), (0, 1), with a duration of 1 time units for each input, and a window of 0.5-time units between each input. We then check whether the output neuron has any spikes. If so, we label that output as 1. If not, it is 0. We then construct a logic gate truth table and assign a logic gate accordingly.

To categorize the non-neuromorphic implementations, each pulse was five time units long, with a five time unit pause between the inputs. We took the value of the output neuron in the middle of the pulse (2.5 time units after input onset). If it was above the threshold of 1.5 units, we considered it 1 else 0. For phase, we considered the circuit on if the maximum value ever exceeded a threshold of 1.5 units for the 40 time unit simulation.

Accuracy testing for ripple carry adder

To test the accuracy and robustness against noise, we inject all neurons in the 4-neuron circuit (E1, E2, I1, I2) with a Gaussian white noise. The mean of the white noise is always zero. We sweep exponentially through increasing standard deviation strength. For the absolute noise, we sweep from a standard deviation of 0.01 nA to a standard deviation of 1.0 nA. For the relative noise condition, we sweep from a standard deviation of 0.1% of the maximal bias of each neuron to 100% of the maximal bias of each neuron. We present 1000 randomly chosen pairs of 4-bit numbers, and run ten trials (each a different Gaussian noise). If the adder returns the correct result for all pairs of numbers and trials—that is 100% accuracy, we consider it robust.

Directional selectivity

To test the phase NIMP logic gate's ability to flexibly respond to directional stimuli, we build a 1D array 20 pixels long. We generate a sinusoidal traveling wave within this 1D pixel array. The luminosity *p* of each pixel is a sinusoidal function of pixel location *x* and *t* in milliseconds. The equation is given by

$$p(x, t) = A\sin(\pi kx - \pi wt) \tag{16}$$

with $k = \frac{1}{10}$, $w = \frac{1}{400}$, and $A = 1.0$. Each CRIREL circuit is connected to adjacent pixels *x* and $x + 1$, such that input currents $in_1(t) = p(x, t)$ (red) and $in_2(t) = p(x + 1, t)$ (green). As this is just a proof-of-concept, we only plot the CRIREL circuit at pixel $x = 10$ and 11 (blue) as well as $x = 15$ and $x = 16$ (yellow).

In order to generate NIMP Logic in the phase mode, we used the Izhikevich version of the phase logic. We used the bias currents in Supplementary Table 12 for both leftward and rightward detection. Note that the different direction selectivities are antisymmetric. This allows a CRIREL sensitive to leftward motion to be reconfigured to rightward detection simply by changing the bias currents.

To challenge the network more, we generate an "object" moving against the sinusoidal traveling wave. To implement this, we use the following equation for pixel luminosity

$$p(x, t) = A\sin(kx - wt) + BU(k_2x - w_2t + 5)\sin(-\pi k_2 x + \pi w_2 t) \tag{17}$$

Here, *U* is the Unit Box function with $U(y) = 1$ if $|y| < \frac{1}{2}$ else 0. We left *A*, *k*, and *w* the same as above. We set $B = -4$, $k_2 = \frac{1}{5}$ and $w_2 = \frac{1}{200}$. This generates a five-pixel wide object moving at the opposite speed to the traveling wave.

In order to detect an object moving against the flow, we use a single CRIREL circuit in our array. For demonstration purposes, we put the CRIREL circuit at pixels 5 and 6. Depending on the bias current chosen, the CRIREL circuit is sensitive to leftward or rightward motion. If the panning camera is moving leftward (inducing a rightward flow), the CRIREL circuit can be set leftward to detect an object moving against the flow, or it can be set to rightward detection to detect the flow. The CRIREL can be reconfigured to do whichever computation the user desires.

Firing rate models

The network structure is given as

$$\tau \frac{de_1}{dt} = -e_1 + f(g_{ee}e_2 - g_{ie}i_1 + b_e) \tag{18}$$

$$\tau \frac{de_2}{dt} = -e_2 + f(g_{ee}e_1 - g_{ie}i_2 + b_e) \tag{19}$$

$$\tau \frac{di_1}{dt} = -i_1 + f(g_{ei}e_1 - g_{ii}i_2 + b_i) \tag{20}$$

$$\tau \frac{di_2}{dt} = -i_2 + f(g_{ei}e_2 - g_{ii}i_1 + b_i) \tag{21}$$

$$\tau \frac{do_1}{dt} = -o_1 + f(g_o(e_1 + e_2)) + b_o) \tag{22}$$

Where e_1 and e_2 are state variables akin to the excitatory neurons. Likewise, i_1 and i_2 are state variables akin to the inhibitory neurons. Here, we set $b_o = 0$. Here b_e and b_i are the bias currents we manipulate to change gates. We set $\tau = 0.25$ We offset the initial conditions from zero. We set $e_1(0) = 0.1$, $e_2(0) = 0.\ 1$, $i_1(0) = 0.02$, $i_2(0) = 0.02$

The activation function for the sigmoidal version was

$$f(x) = \text{sig}(x) = \frac{2}{1 + e^{-2x}} \tag{23}$$

We set the "synaptic" weights as $g_{ee} = 1.5$, $g_{ii} = g_{ei} = g_{ie} = 2$, and $g_o = 1$.

The activation for the capped ReLU function was

$$f(x) = c\text{ReLU}(x) = f(x) = \begin{cases} 0 & \text{if } x{<}0 \\ x & \text{if } 0 \geq x \geq 2 \\ c & \text{if } x \geq 2 \end{cases} \tag{24}$$

We set the "synaptic" weights as $g_{ee} = g_{ii} = 2$, $g_{ei} = g_{ie} = 1.75$, and $g_o = 1$.

Data availability

We have deposited working code into a repository required to recreate the results within this paper at the GitHub link: https://github.com/ajw131/CRIREL_Logic_gates[68]. The code will be made public upon acceptance.

Code availability

The full source code and parameter files are available at https://github.com/ajw131/CRIREL_Logic_gates[68].

Received: 30 April 2024; Accepted: 10 October 2024;
Published online: 03 February 2025

References

1. Lyttle, D. N., Gill, J. P., Shaw, K. M., Thomas, P. J. & Chiel, H. J. Neuromechanical bistability contributes to robust and flexible behavior in a model of motor pattern generation. *BMC Neurosci.* **16**, P33 (2015).
2. Yu, J., Hu, H., Agmon, A. & Svoboda, K. Recruitment of GABAergic interneurons in the barrel cortex during active tactile behavior. *Neuron* **104**, 412–427.e4 (2019).
3. Kato, H. K., Gillet, S. N. & Isaacson, J. S. Flexible sensory representations in auditory cortex driven by behavioral relevance. *Neuron* **88**, 1027–1039 (2015).
4. Wang, T. et al. Reconfigurable neuromorphic memristor network for ultralow-power smart textile electronics. *Nat. Commun.* **13**, 7432 (2022).
5. Kumar, S., Wang, X., Strachan, J. P., Yang, Y. & Lu, W. D. Dynamical memristors for higher-complexity neuromorphic computing. *Nat. Rev. Mater.* **7**, 575–591 (2022).
6. Sun, B. et al. Synaptic devices based neuromorphic computing applications in artificial intelligence. *Mater. Today Phys.* **18**, 100393 (2021).
7. Kagan, B. J. et al. In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. *Neuron* **110**, 3952–3969 (2022).
8. Cai, H. et al. Brain organoid reservoir computing for artificial intelligence. *Nat. Electron.* **6**, 1032–1039 (2023).
9. Liu, B., White, A. J. & Lo, C.-C. Augmenting flexibility: mutual inhibition between inhibitory neurons expands functional diversity. Preprint at *bioRxiv* https://www.biorxiv.org/content/early/2022/01/03/2020.11.08.371179 (2022).
10. Webster-Wood, V. A., Gill, J. P., Thomas, P. J. & Chiel, H. J. Control for multifunctionality: bioinspired control based on feeding in *Aplysia californica*. *Biol. Cybern.* **114**, 557–588 (2020).
11. Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T. & Wang, X.-J. Task representations in neural networks trained to perform many cognitive tasks. *Nat. Neurosci.* **22**, 297–306 (2019).
12. Morton, D. W. & Chiel, H. J. Neural architectures for adaptive behavior. *Trends Neurosci.* **17**, 413–420 (1994).
13. Machens, C. K., Romo, R. & Brody, C. D. Flexible control of mutual inhibition: a neural model of two-interval discrimination. *Science* **307**, 1121–1124 (2005).
14. Lyttle, D. N., Gill, J. P., Shaw, K. M., Thomas, P. J. & Chiel, H. J. Robustness, flexibility, and sensitivity in a multifunctional motor control model. *Biol. Cybern.* **111**, 25–47 (2017).
15. Shenoy, K. V., Kaufman, M. T., Sahani, M. & Churchland, M. M. A dynamical systems view of motor preparation: implications for neural prosthetic system design. *Prog. Brain Res.* **192**, 33–58 (2011).
16. Wang, X.-J. Decision making in recurrent neuronal circuits. *Neuron* **60**, 215–234 (2008).
17. Destexhe, A. & Marder, E. Plasticity in single neuron and circuit computations. *Nature* **431**, 789–795 (2004).
18. Leonard, J. L. Network architectures and circuit function: testing alternative hypotheses in multifunctional networks. *Brain Behav. Evol.* **55**, 248–255 (2000).
19. Marder, E. & Thirumalai, V. Cellular, synaptic and network effects of neuromodulation. *Neural Netw.* **15**, 479–493 (2002).
20. Harris-Warrick, R. M. & Marder, E. Modulation of neural networks for behavior. *Annu. Rev. Neurosci.* **14**, 39–57 (1991).
21. Bargmann, C. I. Beyond the connectome: how neuromodulators shape neural circuits. *Bioessays* **34**, 458–465 (2012).
22. Najafi, F. et al. Excitatory and inhibitory subnetworks are equally selective during decision-making and emerge simultaneously during learning. *Neuron* **105**, 165–179.e8 (2020).
23. Fritz, J., Shamma, S., Elhilali, M. & Klein, D. Rapid task-related plasticity of spectrotemporal receptive fields in primary auditory cortex. *Nat. Neurosci.* **6**, 1216–1223 (2003).
24. Kuchibhotla, K. V. et al. Parallel processing by cortical inhibition enables context-dependent behavior. *Nat. Neurosci.* **20**, 62–71 (2017).
25. Rodgers, C. C. & DeWeese, M. R. Neural correlates of task switching in prefrontal cortex and primary auditory cortex in a novel stimulus selection task for rodents. *Neuron* **82**, 1157–1170 (2014).
26. Niell, C. M. & Stryker, M. P. Modulation of visual responses by behavioral state in mouse visual cortex. *Neuron* **65**, 472–479 (2010).
27. Erzurumlu, R. S. & Gaspar, P. How the barrel cortex became a working model for developmental plasticity: a historical perspective. *J. Neurosci.* **40**, 6460–6473 (2020).
28. Williamson, R. S., Hancock, K. E., Shinn-Cunningham, B. G. & Polley, D. B. Locomotion and task demands differentially modulate thalamic audiovisual processing during active search. *Curr. Biol.* **25**, 1885–1891 (2015).
29. Jaramillo, S., Borges, K. & Zador, A. M. Auditory thalamus and auditory cortex are equally modulated by context during flexible categorization of sounds. *J. Neurosci.* **34**, 5291–5301 (2014).
30. Wang, C.-T., Lee, C.-T., Wang, X.-J. & Lo, C.-C. Top-down modulation on perceptual decision with balanced inhibition through feedforward and feedback inhibitory neurons. *PLoS ONE* **8**, e62379 (2013).
31. Lo, C.-C. & Wang, X.-J. Cortico-basal ganglia circuit mechanism for a decision threshold in reaction time tasks. *Nat. Neurosci.* **9**, 956–963 (2006).
32. Wong, K.-F. & Wang, X.-J. A recurrent network mechanism of time integration in perceptual decisions. *J. Neurosci.* **26**, 1314–1328 (2006).
33. Hennequin, G., Vogels, T. & Gerstner, W. Optimal control of transient dynamics in balanced networks supports generation of complex movements. *Neuron* **82**, 1394–1406 (2014).
34. Ayuso-Martinez, A., Casanueva-Morato, D., Dominguez-Morales, J. P., Jimenez-Fernandez, A. & Jimenez-Moreno, G. Spike-based building blocks for performing logic operations using spiking neural networks on spinnaker. In *2022 International Joint Conference on Neural Networks (IJCNN)* 1–9 (IEEE, 2022).
35. Mo, L. & Wang, M. Logicsnn: a unified spiking neural networks logical operation paradigm. *Electronics* **10**, 2123 (2021).

36. Papernot, N. et al. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)* 372–387 (IEEE, 2016).
37. Galván, E. & Mooney, P. Neuroevolution in deep neural networks: current trends and future challenges. *IEEE Trans. Artif. Intell.* **2**, 476–493 (2021).
38. Helfer, P. et al. Context modulation enables multi-tasking and resource efficiency in liquid state machines. In *Proc. 2023 International Conference on Neuromorphic Systems*, ICONS '23 1–9 (Association for Computing Machinery, 2023).
39. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
40. Itti, L. & Koch, C. Computational modelling of visual attention. *Nat. Rev. Neurosci.* **2**, 194–203 (2001).
41. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci. USA* **79**, 2554–2558 (1982).
42. Izhikevich, E. M. Simple model of spiking neurons. *IEEE Trans. Neural Netw.* **14**, 1569–1572 (2003).
43. Sheridan, P. M., Du, C. & Lu, W. D. Feature extraction using memristor networks. *IEEE Trans. Neural Netw. Learn. Syst.* **27**, 2327–2336 (2015).
44. Ma, W., Zidan, M. A. & Lu, W. D. Neuromorphic computing with memristive devices. *Sci. China Inf. Sci.* **61**, 1–9 (2018).
45. Pan, C. et al. Reconfigurable logic and neuromorphic circuits based on electrically tunable two-dimensional homojunctions. *Nat. Electron.* **3**, 383–390 (2020).
46. Zhang, W. et al. Neuro-inspired computing chips. *Nat. Electron.* **3**, 371–382 (2020).
47. Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Ryu, S. I. & Shenoy, K. V. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron* **68**, 387–400 (2010).
48. Beer, R. D. On the dynamics of small continuous-time recurrent neural networks. *Adapt. Behav.* https://journals.sagepub.com/doi/10.1177/105971239500300405 (1995).
49. Izhikevich, E. M. *Dynamical Systems in Neuroscience* (MIT Press, 2007).
50. Herstel, L. J. & Wierenga, C. J. Network control through coordinated inhibition. *Curr. Opin. Neurobiol.* **67**, 34–41 (2021).
51. Sadeh, S. & Clopath, C. Inhibitory stabilization and cortical computation. *Nat. Rev. Neurosci.* **22**, 21–37 (2021).
52. Mongillo, G., Rumpel, S. & Loewenstein, Y. Inhibitory connectivity defines the realm of excitatory plasticity. *Nat. Neurosci.* **21**, 1463–1470 (2018).
53. Hoppensteadt, F. C. & Izhikevich, E. M. *Weakly Connected Neural Networks* (Springer-Verlag,1997).
54. Hoppensteadt, F. C. & Izhikevich, E. M. In *Weakly Connected Neural Networks* (eds Hoppensteadt, F. C. & Izhikevich, E. M.) Ch. 1–4, 11 (Springer, 1997).
55. Singh, A. K. *Digital VLSI Design* (Prentice Hall, 2010).
56. Borst, A. & Helmstaedter, M. Common circuit design in fly and mammalian motion vision. *Nat. Neurosci.* **18**, 1067–1076 (2015).
57. Lo, C.-C., Wang, C.-T. & Wang, X.-J. Speed-accuracy tradeoff by a control signal with balanced excitation and inhibition. *J. Neurophysiol.* **114**, 650–661 (2015).
58. Durstewitz, D., Seamans, J. K. & Sejnowski, T. J. Neurocomputational models of working memory. *Nat. Neurosci.* **3**, 1184–1191 (2000).
59. Ferguson, K. A. & Cardin, J. A. Mechanisms underlying gain modulation in the cortex. *Nat. Rev. Neurosci.* **21**, 80–92 (2020).
60. Davies, M. et al. Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* **38**, 82–99 (2018).
61. Davies, M. et al. Advancing neuromorphic computing with loihi: a survey of results and outlook. *Proc. IEEE* **109**, 911–934 (2021).
62. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
63. Fukuda, T. & Kosaka, T. Ultrastructural study of gap junctions between dendrites of parvalbumin-containing GABAergic neurons in various neocortical areas of the adult rat. *Neuroscience* **120**, 5–20 (2003).
64. Fukuda, T., Kosaka, T., Singer, W. & Galuske, R. A. W. Gap junctions among dendrites of cortical GABAergic neurons establish a dense and widespread intercolumnar network. *J. Neurosci.* **26**, 3434–3443 (2006).
65. Gallego, G. et al. Event-based vision: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 154–180 (2020).
66. Kogo, N. et al. Dynamics of a mutual inhibition circuit between pyramidal neurons compared to human perceptual competition. *J. Neurosci.* **41**, 1251–1264 (2021).
67. Huang, Y.-C. et al. A single-cell level and connectome-derived computational model of the drosophila brain. *Front. Neuroinform.* **12**, 99 (2019).
68. White, A. J. et al. Project title. https://github.com/ajw131/CRIREL_Logic_gates (2024).

Acknowledgements

We would like to thank Dr. Chou P. Hung and Dr. Andre V. Harrison for their feedback on the manuscript. This work is partially supported by the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project, a joint fund from the Ministry of Education (MOE) and the Ministry of Science and Technology (MOST) in Taiwan. This work was also partially supported by the National Science and Technology Council (Taiwan), 113-2218-E-007-019-, and 113-2321-B-002-029-, and 111-2311-B-007-011-MY3.

Author contributions

C.-C.L. and K.-A.W. provided supervision for the entire project B.L. and A.J.W. discovered the logic gates and developed the theory. B.L., A.J.W., and C.-C.L. developed simulations and robustness tests. B.L., M.-J.H., and A.J.W. ran simulations. M.-J.H. developed the ripple adder. M.-F.C. provided critical advice that guided the development of the motion detector and generalized firing rate simulations. A.J.W. and C.-C.L. wrote the manuscript. All Authors have approved of the manuscript.

Competing interests

A.J.W., B.L., and C.-C.L. are inventors in Taiwan Patent No. #I725914 and United States Patent No. #US12079709B2 related to the research presented in this paper. Additionally, A.J.W., B.L., and C.-C.L. have received royalty payments from Egis Technology Inc. for granting an exclusive license of these patents. The licensing arrangement has been reviewed and approved by The National Tsing Hua University in accordance with its policy on the management and commercialization of intellectual properties. The remaining authors declare no competing interest.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s44335-024-00016-y.

Correspondence and requests for materials should be addressed to Alexander James White or Chung-Chuan Lo.