Evolving Artificial Spin Ice for Robust Computation

ARTHUR PENTY*, GUNNAR TUFTE

Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

Artificial spin ice is a magnetic metamaterial showing particular promise as a novel substrate for unconventional computing. While simulations are invaluable for investigating new computational substrates, results must be robust to the noise and disorder of the physical world for device realisation. Here we investigate the computational robustness of artificial spin ice towards fabrication disorder. Using an evolutionary search, we explore different geometries of artificial spin ice for robust computation. We show that by neglecting to consider disorder in the search, we obtain geometries that suffer greatly when disorder is introduced. We then demonstrate that by explicitly including disorder as part of the evolutionary search process, we are able to discover novel geometries that are robust against disorder. We also find that these geometries perform well on new instances of disorder, and when they fail, we see signs of graceful degradation.

Key words: Material Computation, Unconventional Computing, Artificial Spin Ice, Dynamical Systems, Robust Computation, Evolutionary Algorithms, Novelty Search

1 INTRODUCTION

In the welcoming realm of simulation, where all behaves as its theory dictates, it is often easy to design perfect solutions, machines, components; which function exactly as desired. But, as we move our ideas from theory

* email: arthur.penty@ntnu.no

to practice, we are met by the cold hard world of reality, which is one of manufacturing defects, thermal noise and faulty components. These aberrations, discrepancies between the theoretical scenario and the physical actuality, lead to variation or *disorder* in the otherwise identical components.

For computing devices to be useful they must have some level of *robustness*, a resistance or invariance to the disorder of the physical world. In classical computing, disorder and noise is remedied with the digital abstraction, thresholding values to either 0 or 1. While this increases the robustness of these devices it comes at the cost of increased resources and thus vastly decreases its computational efficiency [21]. Further robustness can be gained through the use of checksums or replicating components and aggregating their output through voting [19]. Both of these add redundancy and require addition processing, further reducing the computational efficiency.

In contrast to the top-down style of choosing a computing paradigm and imposing it on your substrate of choice, material computation [27] tries to first discover useful phenomena naturally present in a substrate and exploit them directly for computation. As a result, we do not incur the extra resource cost of layering on abstraction levels to restrain the computation.

A particularly promising substrate is artificial spin ice (ASI) [25]. ASI is a magnetic metamaterial displaying a wide range of emergent behaviour, including ferromagnetic and anti-ferromagnetic ordering [26] and phase transitions [2, 18] . Its rich behaviour, in conjunction with the extensive tunabilty, makes it a prime candidate for material computation [8, 11, 15]. If this computation is to be harnessed for a ASI-based device, then it must be robust to the disorder accrued in the fabrication process.

This paper is organised as follows. Section 2 gives an introduction to ASI, their use in computation, and how disorder may arise in them. Section 3 briefly outlines the classical approach to managing disorder in the field of engineering and how this contrasts with the philosophy of material computation. In section 4 we detail how we evaluate and evolve ASI geometries. First, we perform a novelty search without disorder and then evaluate the effects of disorder on the discovered geometries. Secondly, we perform a novelty search where disorder is considered as part of the search. Section 5 contains the results and analysis of the two evolutionary searches. Finally, in section 6 we make our concluding remarks.

2 ARTIFICIAL SPIN ICE

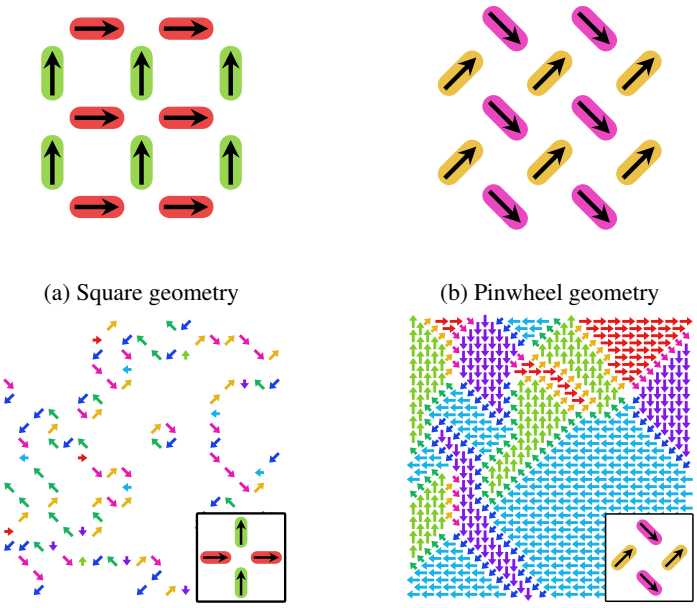
ASI is a substrate consisting of interacting nanomagnets [25]. The state of an ASI is the collective state of all its constituent nanomagnets. The state of a magnet is determined by the state of its neighbours and any other external stimuli.

The magnets in an ASI are mono-domain and bi-stable. When the magnets are sufficiently small, it is always energetically favourable for the internal spins of a magnet to be aligned, meaning we can consider each magnet as a single spin vector. Furthermore, the shape of the magnet is elongated such that this spin vector will always align along this stretched axis. Together these two properties allow us to consider the artificial spin of each magnet to be a binary variable expressing which direction the aggregate of the internal spins points along the stretched axis.

The artificial spin of a magnet can change or 'flip' given sufficient encouragement from its neighbouring magnets, an externally applied magnetic field, or stochastically due to temperature. The coercive field h_k of a magnet is the minimum amount of magnetic field felt by the magnet that will cause it to flip. One view of this flipping is to think of the magnet as 'wanting' to be in the lowest energy state as possible. It achieves this by aligning itself as much as possible with the sum of the magnetic fields acting on it, providing it has the required energy to do so.

ASI systems display exotic properties when considered as a metamaterial. A metamaterial consists of units or building blocks, analogous to the atoms in a bulk material. Using this analogy, we can consider the 'material' properties arising from the configurations of these units. A metamaterial perspective of ASI zooms out from the individual magnets, and looks at the large scale patterns and phenomena similar to that of a bulk material. The difference being that these phenomena and material properties can be tuned easily by altering the configuration of the constituent magnets.

The specific arrangement and orientations of the magnets in an ASI is referred to as the ASI *geometry*. Changing the geometry of an ASI has profound effects on its large scale metamaterial properties. As an example we can consider the Square ASI geometry shown in Fig. 1a, one of the most simple and well-studied ASI geometries. The Square ASI exhibits anti-ferromagnetic behaviour, meaning the spins in the system tend towards lining up in a chequerboard pattern causing net demagnetised regions, shown as large white regions in Fig. 1c. If we make a small adjustment to the Square geometry, rotating every magnet by 45°, we obtain the Pinwheel geometry (Fig. 1b). Now, the



(a) Square geometry

(b) Pinwheel geometry

(c) Square emergence

(d) Pinwheel emergence

FIGURE 1: (a) & (b) Examples of well-studied ASI geometries. (c) & (d) Corresponding emergent phenomena associated with each of the geometries. Here spins have been averaged into a single spin per vertex to make the emergent properties clearly visible. An inset in the bottom right corner shows the definition of a vertex in the two geometries. White regions indicate a net magnetisation of zero, i.e., the spins cancel out.

magnets self-organise into ferromagnetic patterns, expressed by large regions of coherent magnetisation, shown as blocks of uniform colour in Fig. 1d). These are just two examples of the wealth of emergent phenomena that can be achieved through tuning the ASI geometry.

2.1 Computing with ASI

ASI is an increasingly promising substrate for computation [7, 8, 15]. It is of particular interest for material computation, due to exhibiting self-organisation and rich emergent behaviour arising from its simple binary elements. For computation, we consider the ASI as a dynamical system. It can be perturbed with magnetic fields as a means of providing input to the system. An output

from the system can be obtained by reading off the states of the magnets in the system, or by some aggregation of these states.

Following this methodology, Reservoir Computing (RC) [9, 20] is an obvious and natural choice for a suitable computational paradigm. RC takes some candidate computational substrate, perturbs it with input, then applies a linear readout layer to the observable state of the system to perform some function or task. Tasks include classification, regression and time series prediction [23, 29]. Often the readout is applied frequently as the substrate is being perturbed, though it is also possible to take a one-shot approach and only measure after the full input sequence [1, 6]. As the linear readout has no memory or ability to compute non-linear functions, any memory or non-linearity required to solve the particular computational task must arise from the substrate.

To perform computation, RC relies on the substrate to provide a rich response to input. Hence, the substrate must be capable of rich dynamics, and specifically it must have rich dynamics under the chosen input encoding. For instance, if you have a classification problem where you want to group inputs into 10 different bins, but the substrate is only capable of terminating in 5 different states, then clearly you cannot solve the problem [3]. Similarly, if your system is producing a very high number of different output states, it may be too sensitive to noise. It is then of interest to be able to tune the number of states a substrate produces, or even investigate that it is capable of producing 'enough' unique responses to input.

As a rich response to input is crucial for material computing, it is useful to have a means of quantifying the behaviour of a substrate to determine its suitability. One way to quantify and categorise the dynamics of an ASI is to count the number of different states or 'outputs' produced by the ASI, as it responds to different inputs. Jensen et al. [12] use this state count measure to investigate how the dynamics of an ASI change with respect to the strength of an external global field, acting as the input to the system.

In our previous work, we showed how the geometry of an ASI can be tuned through use of an Evolutionary Algorithm (EA) to obtain different number of unique states visited over the trajectory of the ASI [22]. Though such studies have focused on investigating the ASI for computation through use of simulation, the goal is computation *in materio*.

2.2 ASI Manufacture

Disorder in an ASI is accumulated in the manufacturing process. An example of a typical process is as follows: ASI structures are fabricated using a

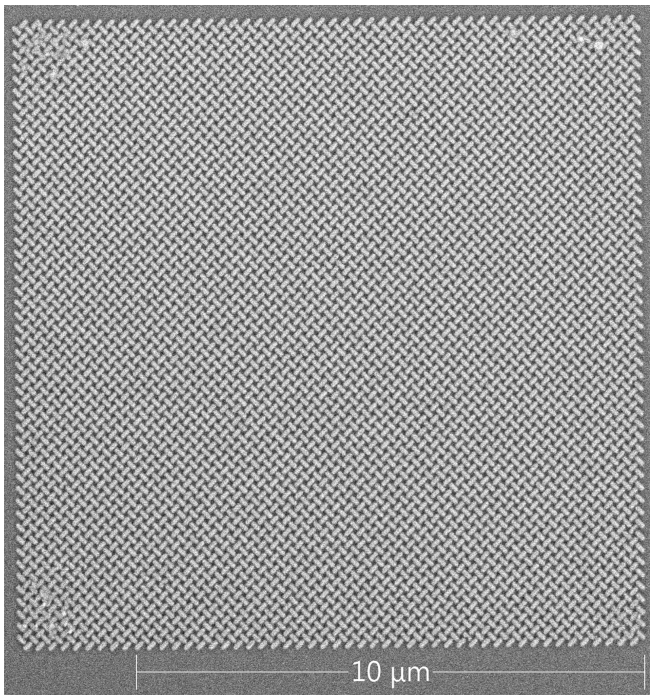


FIGURE 2: A scanning electron microscope image of a Pinwheel ASI (Fig. 1b) produced at the the NTNU NanoLab for a recent experiment on clocking protocols in ASI [14]

lift-off electron beam lithography (EBL) process. First a resist mixture is prepared on a wafer and the desired nanomagnetic pattern is transferred onto the resist through EBL exposure. Permalloy ($Ni_{80}Fe_{20}$) is then deposited onto the wafer though electron beam evaporation, followed by a thin coating of aluminium to prevent oxidation. The final step is lift-off where a chemical removal agent and ultrasonic agitation is used to strip off all but the nanomagnetic islands. An example of an ASI fabricated in this way can be seen in Fig. 2.

Disorder can arise due to material impurity, the finite precision of the EBL or through the sample coming into contact with contaminants through handling or otherwise. In this instance disorder presents itself as a slight variance on the size, shape or composition of each nanomagnet. Though small, these alterations can have a significant impact on the behaviour of the metamaterial.

3 DISORDER AND ROBUST COMPUTATION

Disorder can be found both within a system, such as the timing variations of transistors within the same chip [4], as well as between systems, e.g., devices produced in the same production batch having slightly different performance. Given that any physical device must realise its computation through the operating of physical components, disorder in these physical components may adversely affect the computation.

A common approach to dealing with disorder in the fields of engineering and conventional computer science, is to minimise disorder to the point at which it has no meaningful effect on the behaviour of the system. This is achieved by working within certain tolerances and pushing these tolerances with ever more complex, highly specialised and sensitive manufacturing processes, e.g., developments in lithography [5].

In material computation, importance is placed on not constraining your substrate to conform to some preconceived computational model [27]. Ideally it should be possible to configure or interact with the substrate in such a way that the effects of disorder are reduced. This is preferable to an engineered solution, which may add additional overhead or restrict the behaviour of the substrate.

Alternatively disorder need not be assumed a hindrance. In fact, it has been shown that in some non-linear dynamical systems, both electronic and biological, a certain amount of noise can improve sensitivity [28]. In this sense, disorder can be seen as just another phenomenon to exploit with material computing.

We refer to an object's ability to maintain performance in an imperfect or unexpected environment as its robustness. Evolutionary Robotics, another field where simulations are often a necessary precursor to physical demonstration, uses the concept of a reality gap to describe how that which was developed in simulation may not function correctly when translated into the physical world. Jakobi et al. [10] show that a robotic controller trained in simulation can perform better when controlling a real robot if noise is added to the simulation, thus reducing the reality gap.

4 METHODS

4.1 State Count Metric

For our evaluation of dynamics, we use a length n sequence of bits as input. In our input encoding, each bit corresponds to perturbing an ASI, with one

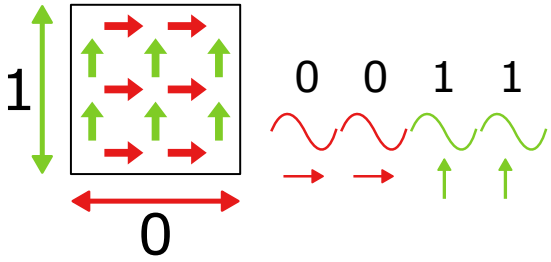


FIGURE 3: Input encoding: input bit strings are converted into a time series of field applications. Each field application follows one full period of a sine wave. The bit determines the angle at which the field is applied relative to the ASI, where zero is an angle of 0° and one is an angle of 90°.

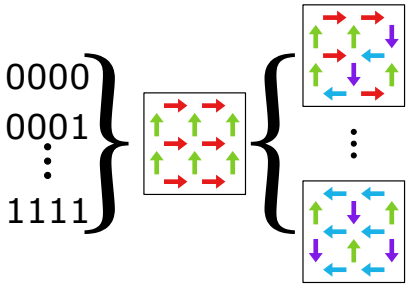


FIGURE 4: Through applying different inputs to an ASI and recording the final state we can build up a lookup table or truth table to understand the function a given ASI geometry is performing. From this the state count metric can be calculated.

cycle of a sinusoidal global field. 0 is mapped to a cycle with a field angle of 0°, while 1 is mapped to a field angle of 90°. Fig. 3 shows a 4-bit input, and the corresponding series of field applications it is mapped to.

We evaluate the system on every possible length n bit sequence, and record the final state of the system after each sequence (Fig. 4). The state count is then the number of unique final states we observe after applying the different inputs. As we have a finite number of possible inputs which we exhaustively search, the state count measure is the cardinality of the range of the function computed by the ASI.

State count can be used as a descriptor of the intrinsic computational properties of the ASI. For a state count s where $s << n$ we have an n-bin classi-

fier. When $s = n$ we have a system with perfect (likely non-linear) memory, that is, all n-bits of input can be recovered by observing the final state of the system. Between the two extremes there is a mixed regime where the substrate exhibits some classification power and some memory [11].

4.2 flatspin Simulations

We use the flatspin ASI simulator [13] to determine the dynamical properties of an ASI geometry. flatspin allows us to obtain the trajectory of an ASI under a given series of field applications in reasonable time, making evolutionary search feasible.

In flatspin, disorder is expressed as a small random variation on the switching threshold h_k of the magnets in the simulated ASI. The modified switching threshold h'_k is sampled from the normal distribution $\mathcal{N}(h_k, d \cdot h_k)$ where d is a parameter to decide the strength of the disorder. h'_k is sampled independently for each magnet, and remains constant throughout the simulation. In this work we will use either $d = 0\%$ when simulating without disorder, or $d = 4\%$ when simulating with disorder. The value 4% is used as it was found as a rough estimate of the true disorder resulting from our manufacturing process [16].

While the disorder d is kept constant, we can supply different random seeds to flatspin in order to obtain different samples of switching thresholds. We will refer to these different samples as different instances of disorder. Between the different instances, the overall disorder in an ASI will remain similar but will be distributed differently throughout the system.

4.3 Evolving ASI Geometries

Previously, we have set out a representation and methodology for evolving ASI geometries [22], which we will make use of here. Starting from a single magnet, a generative, deterministic algorithm uses a set of tiles to build a geometry. In this way the same geometry can be grown to different sizes by varying the number of iterations for which the growth process is run.

Within our framework, an ASI geometry is represented as an ordered list of 2-magnet tiles. Each tile contains one *origin magnet* that is always at the centre of the tile and 0°rotation (w.r.t the tile), and a second *free magnet* that can be anywhere in the tile and have any rotation. The tiles can be thought of as stencils, we select a tile and position and rotate it such that the origin magnet perfectly overlaps a magnet already in the system. We can then 'draw' a magnet with the position and rotation of the free magnet. To generate a geometry, we start with one magnet, apply each tile in order to any magnets

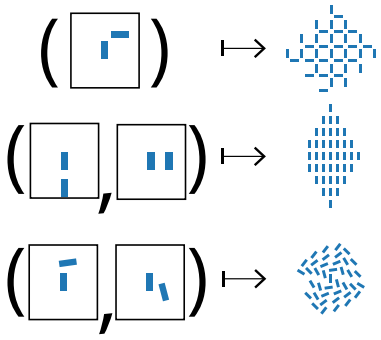


FIGURE 5: Three example tile-sets and the corresponding geometry they produce. The top tile-set produces the Square ASI geometry, the middle tile-set produces the Ising ASI geometry and the bottom tile-set produces a novel geometry.

in the system to create a set of new magnets. Each new magnet is added to the geometry, provided its distance to its nearest neighbour is within a given tolerance. This process is repeated until either the desired number of magnets is reached, or no more magnets can be placed and the geometry converges. Full pseudocode of this process and the tile application process can be found in [22].

Fig. 5 shows some examples of tile sets and the corresponding geometries they produce. We can see that this method can not only reproduce classically studied structures such as the Square and Ising geometry, but also create novel and complex geometries with more organic structures. Further examples of the structures that can be produced and animations showing how they are built can be seen at https://s.ntnu.no/comet-papers-ijuc2023.

By specifying the maximum number of tiles for each geometry in the EA, the complexity of the produced geometries can be controlled. Geometries with more tiles can produce a greater variety of designs, but it comes at the cost of increasing the difficulty of optimising the geometry, and increases the chance of producing malformed geometries, i.e., the generative process terminates prematurely before the requisite number of magnets are produced.

Additionally, the magnets in a tile can be imbued with symbols to achieve a greater diversity of geometries. For a tile application to occur, the symbol of the tile's origin magnet must match the symbol of the magnet of which the

tile is being applied to. These symbols can be parameterised, such as being affected by the angle of the magnet, which aids in breaking symmetry.

Variation in the EA is achieved through mutation of the tiles, either through adding and deleting tiles from the geometry, or modifying the position and angles of magnets withing a tile. In addition, a crossover randomly selects tiles from two geometries to produce a new geometry.

As our interest is not in forcing ASI into a predetermined mode of computation, but to explore the inherent computational properties of ASI, we do not steer our evolutionary search towards a particular state count. We encourage it to explore a breadth of possibilities in the search space. Practically, this means searching using a novelty search [17], where ASI geometries are rewarded for their novelty.

After variation and crossover creates a number of new individuals, a selection phase occurs to ensure the population size remains constant. We do this by running magnetic simulations of the geometries and evaluating the output to determine its fitness. The population is then reduced by removing the geometries of inferior fitness. As we are using novelty search, the fitness will quantify how novel a given geometry's behaviour is.

The parameters shared between both novelty searches are as follows. The population size is 100 and mutation probability and crossover probability are both 0.4. Each geometry can have a maximum of six tiles and a maximum of six symbols. We parameterise the symbol of a magnet as a function of its current rotation. Specifically, we use a piece-wise function to select the symbol of a magnet depending on whether its orientation is greater or less than 180°. All geometries are grown to have exactly 100 magnets. If a geometry's growth process terminates early, before reaching 100 magnets, it is assigned the worst possible fitness.

4.4 Disorder Free Search

First we consider the disorder free case on 8 bits of input ($n = 8$). In this case the novelty space is small and discrete, so we can use a very simplistic novelty measure. If we let s_i be the state count for an individual i and N be the set of all so far unseen state counts, then the novelty of an individual X is given by:

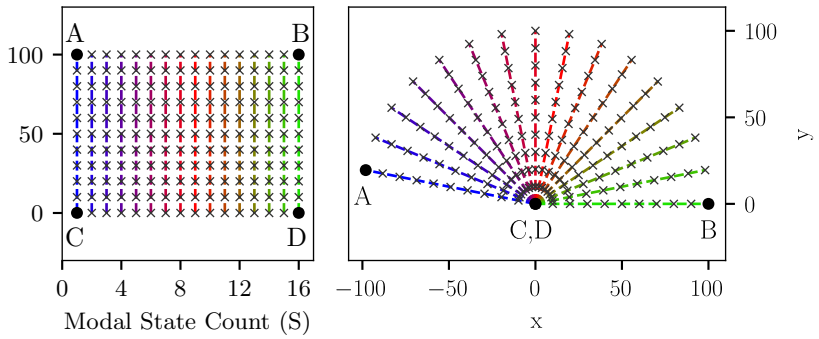
$$\text{novelty}(X) = -\min_{n \in N}(|s_X - n|) \tag{1}$$

i.e., an individual's novelty is how close its state count is to the nearest unobserved state count. Using this fitness function and our ASI representation,

a novelty search was executed in search of state counts from 2 to 255 (state count 1 was excluded as it is not of computational value and hindered the search).

To assess the effects of disorder we take the ASI geometries discovered in the disorder free search and evaluate their performance when simulated with disorder. For each of the discovered state counts we take a geometry with that state count and run 100 independent simulations with different instances of disorder and record in what percentage of these it retains its original state count.

4.5 Novelty Search with Disorder



(a) (b)

FIGURE 6: (a) Example of points distributed evenly in the S-r search space, and four extreme points A, B, C and D. (b) The points from (a) after being transformed into the x-y space by the transformation given in Eq. (2), illustrating how the transformation affects the proximity of points and thereby the novelty metric.

Next, disorder is introduced into the search process to assess the capability of the EA to produce geometries with robust behaviour. To evaluate the robustness of a geometry it is insufficient to test on just one instance of disorder. The number of different disorder instances to test on must be enough that the EA does not over-fit and learn to exploit phenomena specific to the particular instances of disorder. Thus the state count of a geometry is no longer a single value but an ensemble of different state counts and the frequency at which they occur.

We evaluate each geometry on 100 different instances of disorder. This increases the run time of the novelty search 100 times. In order to run the search in feasible time we decrease the number of input bits to $n = 4$, reducing the number of state counts to 16.

Here, we define the modal state count of a geometry S as the state count observed most frequently over the different disorder instances (ties broken arbitrarily). We then define the geometry's robustness r as the proportion of observations where the state count was S.

Our search space for novelty search is the 2D space spanned by S and r. The canonical way to define novelty of an individual is to use some metric on the population of individuals in its behavioural neighbourhood, i.e., the amount or their proximity. Instead of doing this in the S-r space, we first transform the coordinates into a new x-y space with the transformation:

$$\begin{aligned} x &= r\cos(\theta) \\ y &= r\sin(\theta), \\ \text{where } \theta &= (1 - \frac{S}{S_{\text{max}}})\pi \end{aligned} \tag{2}$$

The effect of the transformation is to use the S and r as a form of polar coordinates, as seen in Fig. 6. Applying the transformation before evaluating the novelty of individuals give two main benefits: firstly, prior to the transformation individuals with less than 50% robustness make up for 50% of the search space. After the transformation these less robust individuals are compressed together taking up 25% of the search space.

Secondly, consider 4 individuals, A, B, C and D (shown in Fig. 6), where A and B have high robustness but with maximally different state counts, and C and D have low robustness and maximally different state counts. We can see from Fig. 6a that these individuals constitute the 4 corners of the S-r space, thus the behavioural distance, dist(A, B), is equal to the distance between C and D, dist(C, D). However, individuals with low robustness produce a wide variety of state counts. Hence, the modal state count metric is a poor discriminator for low robustness individuals. As such we consider C and D to be more behaviourally similar than A and B. After the transformation, we can see in Fig. 6b that dist(C, D) is now minimised and dist(A, B) is maximised. Thus, post transformation, the novelty metric more accurately portrays our understanding of novelty in the behaviour of the individuals.

To ensure the geometries are truly robust to disorder and are not just performing well on the specific instances of disorder they are tested on, we evaluate the most robust individual for each state count against a test-set of 100

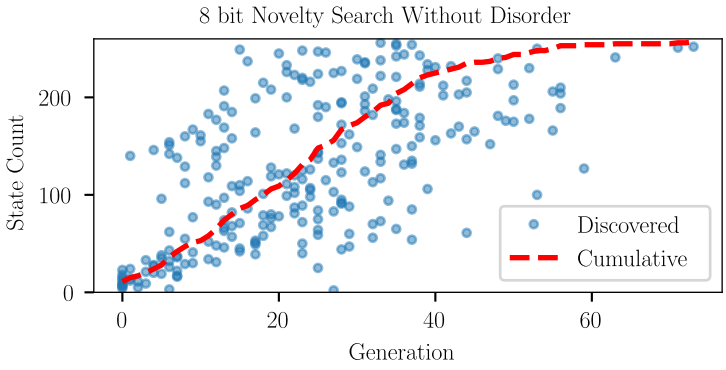


FIGURE 7: Results of novelty search to find geometries of different state counts. The blue dots show in which generation a state count first observed. The red dashed line shows the total number of different state counts observed by that generation in in the evolutionary run.

new, unseen instances of disorder.

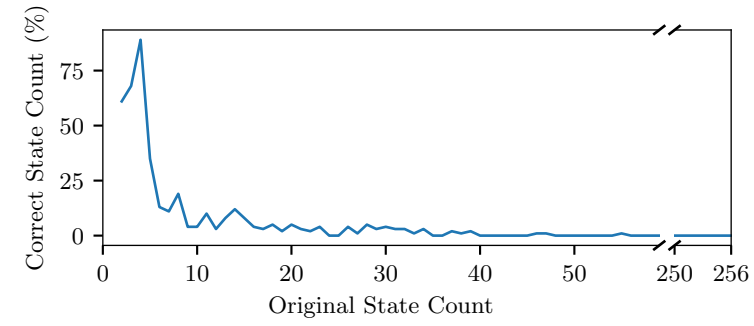
5 RESULTS

First we report the results of the search without disorder with an 8-bit inputs (section 4.4) and the consequences of adding disorder to the resulting geometries. Secondly, we present the result of the evolutionary search where disorder was considered from the start and inputs of 4-bits were used (section 4.5).

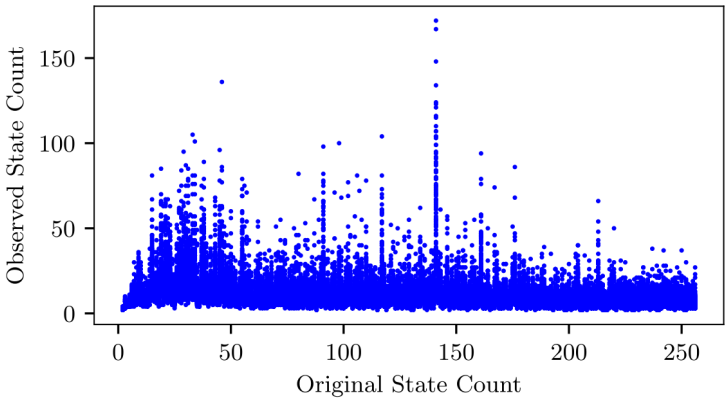
5.1 Results of the Disorder Free Search

Fig. 7 shows the state counts discovered each generation during the evolutionary run. All 254 unique state counts were discovered by generation 73. We see stable discovery of new state counts for the majority of the run which tapers off towards the end, perhaps as the remaining state counts are harder to achieve, but also as there are fewer left to find.

From the results shown in Fig. 8a, we can see all but the first few state counts suffer significantly when disorder is introduced. The highest performance is the geometry that was discovered with state count of 4 which retained its state count in 84% of the disorder instances. All geometries with an original state count of 16 or higher retained their state count in 5% or less of



(a)



(b)

FIGURE 8: The results of taking the geometries discovered in the disorder free search, and evaluating them with disorder. One geometry for each of the 254 state counts discovered was evaluated against 100 different disorder instances. The original state count is the state count observed when no disorder is present. (a) shows the percentage of the separate disorder instances in which the geometry still exhibits its original state count. All geometries with original state counts above 55 have a correct state count of 0%. (b) shows the original state count of a geometry versus all the different state counts that were observed when that geometry was subject to the different instances of disorder.

the disorder instances, and those with original state counts above 55 did not retain it in any of the disorder instances.

These results show the need to take disorder into consideration when designing computational ASI geometries for the real world. Had we wanted to fabricate one of these geometries we would have at best an 84% success rate if we choose a state count of 4 where the robustness was highest. If we wanted to fabricate one of the high state count geometries then there would be almost no chance of success.

Fig. 8b shows the distribution of the state counts each geometry produces when simulated with the different instances of disorder. We can see that there is a lot of spread in the observed state counts for each geometry. Also, there is no apparent trend or relationship between the original state count of a geometry and the state counts observed when it is subjected to disorder. This strongly indicates that the computational properties discovered with the novelty search have been destroyed by the application of disorder.

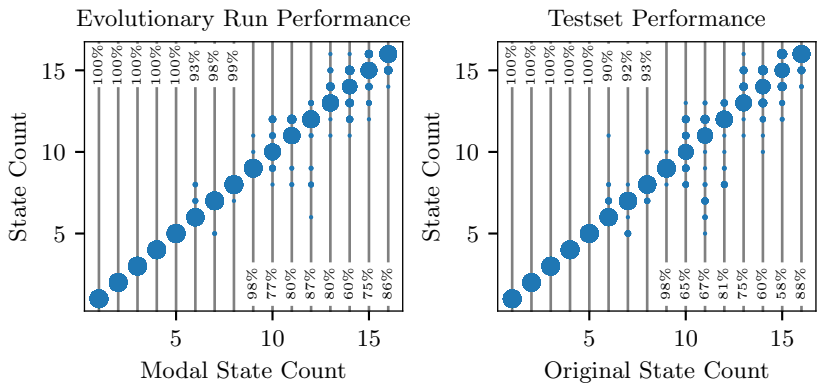
5.2 Results of the Evolutionary Search with Disorder

Fig. 9a shows the most robust individual found for each state count, and the distribution of state counts it produces. Immediately, we can see all state counts were found with some level of robustness, and that for the first 5 state counts an individual was found with perfect robustness, i.e., it produced the same state count over all the different instances of disorder.

From Fig. 9b we see that all the individuals with 100% robustness maintained this level of robustness on the test-set, a strong indication that these individuals are truly robust to this level of disorder. On the other individuals, we mostly see a reduction in robustness in the range $3\% - 13\%$ with two exceptions being a reduction of 17% and an increase of 2%.

It can also be seen in Fig. 9b that, where the robustness is not 100%, the other observed state counts are clustered around the modal state count. This shows that there exists locality in the state count of the ASI, indicating there is some gradient in the search space. This locality in the state count could be viewed as kind of resilience or graceful degradation, where the system is not performing optimally but still retains some level of quality. Furthermore, in applications such as RC, it may not be necessary to achieve an exact state count, and rather it is desirable to be within a particular region of the search space which corresponds to some desired dynamical regime. In such a situation, these results show particular promise.

Fig. 10 shows how the robustness for each state count improves over the course of the novelty search. We can see that as in the no disorder case



(a) (b)

FIGURE 9: (a) The spread of observed state counts in the most robust geometry for each modal state count, tested on the set of disorder instances used in the evolutionary run. The size of the the blue dots is relative to how many times a certain state count was observed over the different disorder instances. Each column is labelled with the corresponding geometry's percentage robustness. (b) The same geometries but simulated on 100 new unseen instances of disorder. Here the x-axis shows the modal state count the geometry produced on the original set of disorder instances.

(Fig. 7), smaller state counts were found earlier. Moreover, it was seemingly much easier to find fully robust candidates at the lower state counts, with the first five being discovered almost immediately. In the higher state counts there are clearly some regions of step-wise improvement as newly discovered individuals are iterated on and improved, though in some cases there are also large stagnant periods with no improvement.

A selection of the most robust geometries for different state counts is shown in Fig. 11. The geometry with state count 15 stands out amongst the others as appearing particularly 1-dimensional (most of its extent is along one direction). One could hypothesise that a more 1-dimensional geometry would lend itself to producing higher state counts or having lower robustness, as each magnet has fewer neighbours and thus more freedom to 'choose' its own state. However, taking the dataset of the full evolutionary run we were not able to find any significant correlations to corroborate such a hypothesis. It may be that such correlations exist, and that this dataset was not sufficient

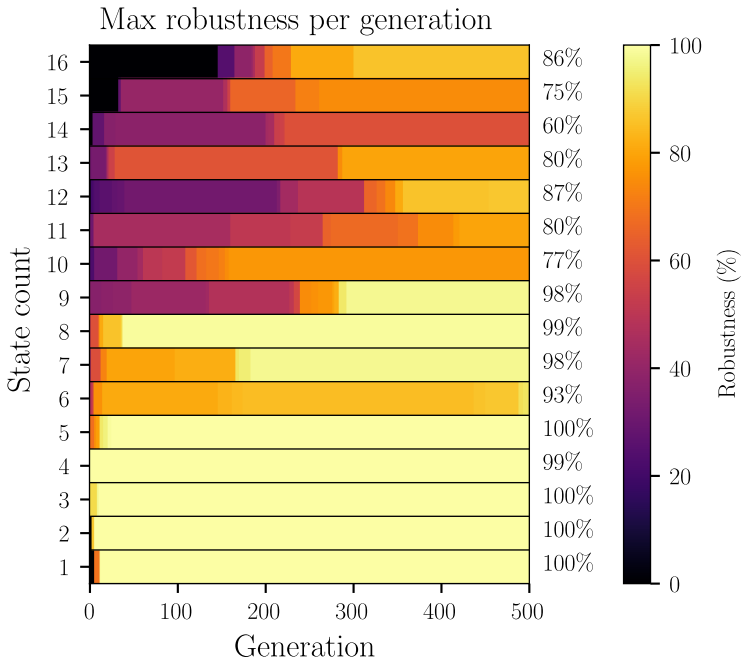


FIGURE 10: The evolution of robustness in each state count over the course of the evolutionary run. The colour indicates the highest robustness individual for the given state count observed so far in the evolutionary run (the individual may not still be present in the population). The maximum robustness achieved by the end of the run is shown on the right.

to observe them.

6 CONCLUSION

In this work we have shown the need to consider disorder when investigating a metamaterial for computation, and demonstrated the extent to which disorder can destroy computational properties observed in a simulated ASI. We have also shown that by including disorder in our evolutionary search, we achieve a substantial increase in the robustness of the produced ASI geometries across all state counts.

In addition to finding geometries with 100% robustness for some state counts, we have also shown that the geometries which were not fully robust to disorder did show strong signs of resilience or graceful degradation where the

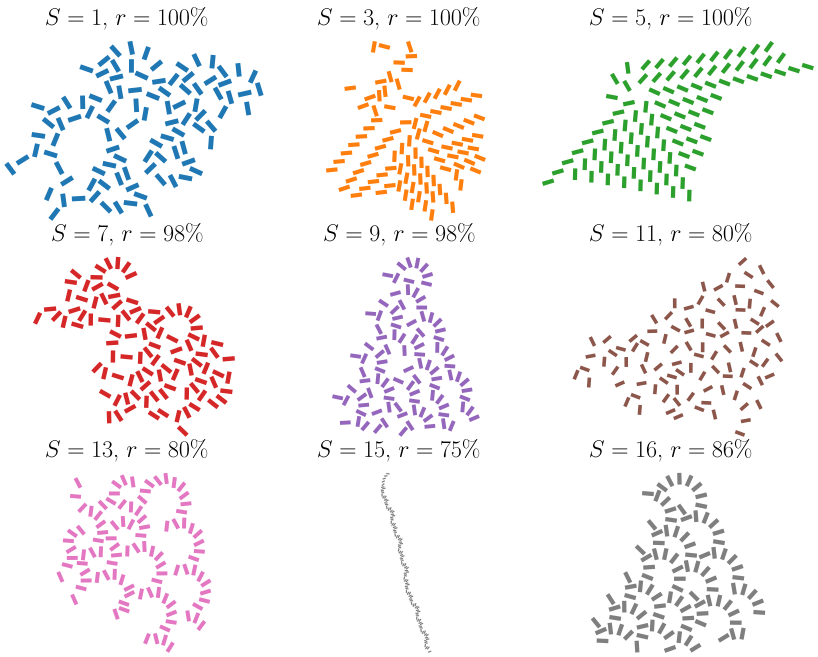


FIGURE 11: A selection of geometries with the highest robustness (r) for their state count (S). Each geometry was grown until it contained exactly 100 magnets.

state counts were mostly tightly clustered around their modal state count. In applications such as RC where one often wants to target a certain dynamical regime, the state count gives a heuristic to measure or target different regimes. In this case, these results are particularly promising as being close to a certain modal state count is likely sufficient.

Though we have focused on ASI as our substrate for material computation, concepts such as state count and robustness to disorder are widely applicable across different metamaterials and different realisations of material computation. While simulation provides powerful insight into new substrates for computation, any phenomena exploitable for computation in simulation must be robust to disorder if the computation is to achieve the ultimate goal of material computing, computing *in materio*.

7 ACKNOWLEDGEMENTS

This work was funded in part by the Norwegian Research Council TEKNOKONVERGENS project SPrINTER (Grant no. 331821), and in part by the EU FET-Open RIA project SpinENGINE (Grant no. 861618). flatspin [13] simulations were executed on the NTNU EPIC compute cluster [24].

The Research Council of Norway is acknowledged for the support to the Norwegian Micro- and Nano-Fabrication Facility, NorFab, project number 295864

REFERENCES

[1] Luís A Alexandre, Mark J Embrechts, and Jonathan Linton. (2009). Benchmarking reservoir computing on time-independent classification tasks. In *2009 International Joint Conference on Neural Networks*, pages 89–93. IEEE.

[2] L. Anghinolfi, H. Luetkens, J. Perron, M. G. Flokstra, O. Sendetskyi, A. Suter, T. Prokscha, P. M. Derlet, S. L. Lee, and L. J. Heyderman. (November 2015). Thermodynamic phase transitions in a frustrated magnetic metamaterial. *Nature Communications*, 6(1):8278.

[3] William Ross Ashby. (1960). *Design for a Brain: The Origin of Adaptive Behavior*. Chapman & Hall.

[4] Shekhar Borkar. (2005). Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *IEEE Micro*, 25(6):10–16.

[5] Martin Burkhardt, Anuja De Silva, Jennifer Church, Luciana Meli, Chris Robinson, and Nelson Felix. (2019). Investigation of mask absorber induced image shift in euv lithography. In *Extreme Ultraviolet (EUV) Lithography X*, volume 10957, pages 232–250. SPIE.

[6] Mark J Embrechts, Luís A Alexandre, and Jonathan D Linton. (2009). Reservoir computing for static pattern recognition. In *ESANN*.

[7] Jack C Gartside, Kilian D Stenning, Alex Vanstone, Holly H Holder, Daan M Arroo, Troy Dion, Francesco Caravelli, Hidekazu Kurebayashi, and Will R Branford. (2022). Reconfigurable training and reservoir computing in an artificial spin-vortex ice via spin-wave fingerprinting. *Nature Nanotechnology*, 17(5):460–469.

[8] Kwan Hon, Yuki Kuwabiraki, Minori Goto, Ryoichi Nakatani, Yoshishige Suzuki, and Hikaru Nomura. (2021). Numerical simulation of artificial spin ice for reservoir computing. *Applied Physics Express*, 14(3):033001.

[9] Herbert Jaeger. (2001). The "echo state" approach to analysing and training recurrent neural networks - with an Erratum note. Technical Report 148, German National Research Center for Information Technology.

[10] Nick Jakobi, Phil Husbands, and Inman Harvey. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In *Advances in Artificial Life: Third European Conference on Artificial Life Granada, Spain, June 4–6, 1995 Proceedings 3*, pages 704–720. Springer.

[11] Johannes H. Jensen, Erik Folven, and Gunnar Tufte. (2018). Computation in artificial spin ice. In *The 2018 Conference on Artificial Life*, pages 15–22, Tokyo, Japan. MIT Press.

[12] Johannes H. Jensen, Erik Folven, and Gunnar Tufte. (2018). Computation in artificial spin ice. *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, (30):15–22.

[13] Johannes H Jensen, Anders Strømberg, Odd Rune Lykkebø, Arthur Penty, Jonathan Leliaert, Magnus Själander, Erik Folven, and Gunnar Tufte. (2022). flatspin: A large-scale artificial spin ice simulator. *Physical Review B*, 106(6):064408.

[14] Johannes H. Jensen, Anders Strømberg, Ida Breivik, Arthur Penty, Michael Foerster, Miguel Angel Niño, Muhammad Waqas Khaliq, Gunnar Tufte, and Erik Folven. (2023). Clocked dynamics in artificial spin ice. *arXiv:2306.07388 [cond-mat.mes-hall]*.

[15] Johannes H. Jensen and Gunnar Tufte. (2020). Reservoir computing in artificial spin ice. *Artificial Life Conference Proceedings*, (32):376–383.

[16] Amanda Langørgen. (2020). Magnetic force microscopy and micromagnetic simulations of nanoscale magnetic structures and modified artificial spin ices. Master's thesis, NTNU.

[17] Joel Lehman, Kenneth O Stanley, *et al.* (2008). Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336.

[18] Demian Levis, Leticia F. Cugliandolo, Laura Foini, and Marco Tarzia. (May 2013). Thermal Phase Transitions in Artificial Spin Ice. *Physical Review Letters*, 110(20):207206.

[19] Robert E Lyons and Wouter Vanderkulk. (1962). The use of triple-modular redundancy to improve computer reliability. *IBM journal of research and development*, 6(2):200–209.

[20] Wolfgang Maass, Thomas Natschläger, and Henry Markram. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560.

[21] Carver Mead. (1990). Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636.

[22] Arthur Penty and Gunnar Tufte. (07 2021). A Representation of Artificial Spin Ice for Evolutionary Search. *ALIFE 2021: The 2021 Conference on Artificial Life*. 99.

[23] Benjamin Schrauwen, David Verstraeten, and Jan Van Campenhout. (2007). An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007*, pages 471–482.

[24] Magnus Själander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. (December 2019). EPIC: An Energy-Efficient, High-Performance GPGPU Computing Research Infrastructure. *arXiv:1912.05848 [cs]*.

[25] Sandra H. Skjærvø, Christopher H. Marrows, Robert L. Stamps, and Laura J. Heyderman. (January 2020). Advances in artificial spin ice. *Nature Reviews Physics*, 2(1):13–28.

[26] Joseph Sklenar, Yuyang Lao, Alan Albrecht, Justin D. Watts, Cristiano Nisoli, Gia-Wei Chern, and Peter Schiffer. (February 2019). Field-induced phase coexistence in an artificial spin ice. *Nature Physics*, 15(2):191–195.

[27] Susan Stepney. (2008). The neglected pillar of material computation. *Physica D: Nonlinear Phenomena*, 237(9):1157–1164. Novel Computing Paradigms: Quo Vadis?

[28] Kurt Wiesenfeld and Frank Moss. (1995). Stochastic resonance and the benefits of noise: from ice ages to crayfish and squids. *Nature*, 373(6509):33–36.

[29] Francis Wyffels and Benjamin Schrauwen. (2010). A comparative study of reservoir computing strategies for monthly time series prediction. *Neurocomputing*, 73(10-12):1958–1964.