Accelerating recurrent Ising machines in photonic integrated circuits: supplementary material

Mihika Prabhu\textsuperscript{1,*}, †, Charles Roques-Carmes\textsuperscript{1,*}, †, Yichen Shen\textsuperscript{2,4,*}, †, Nicholas Harris\textsuperscript{1,3}, Li Jing\textsuperscript{4}, Jacques Carolan\textsuperscript{1}, Ryan Hamerly\textsuperscript{1,5}, Tom Baehr-Jones\textsuperscript{6}, Michael Hochberg\textsuperscript{6}, Vladimir Čeperić\textsuperscript{4}, John D. Joannopoulos\textsuperscript{2,4,7}, Dirk R. Englund\textsuperscript{1}, and Marin Soljačić\textsuperscript{1,4}

\textsuperscript{1}MIT Research Lab of Electronics, 50 Vassar St, Cambridge, MA 02139, USA
\textsuperscript{2}Lightelligence, 268 Summer Street, Boston, MA 02210, USA
\textsuperscript{3}Lightmatter, 61 Chatham St 5th floor, Boston, MA 02109, USA
\textsuperscript{4}MIT Department of Physics, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
\textsuperscript{5}Nokia Research, 1950 University Ave #600, East Palo Alto, CA 94303
\textsuperscript{6}NTT Research, 171 Madison Ave 1100, New York, NY 10016, USA
\textsuperscript{7}Institute of Soldier Nanotechnologies, 500 Technology Square, Cambridge, MA 02139, USA
* denotes equal contribution
† Corresponding authors: mihika@mit.edu, chrc@mit.edu, ycshen@mit.edu

Published 18 May 2020

This document provides supplementary information to “Accelerating recurrent Ising machines in photonic integrated circuits,” https://doi.org/10.1364/OPTICA.386613, and includes extended details on the experimental configuration, ground state search results, and characterization of phase space exploration.

1. PNP CONFIGURATION SETUP

In this section, we describe the programmable nanophotonic processor (PNP) main control steps. Here, we review the implementation on the PNP of the algorithm already presented in [1]. The PNP configuration can be decomposed in three main steps:

- **General Initialization**: Generates an Ising model and initializes the different inputs to the PNP: algorithm parameters, initial spin state, homodyne detection matrices, etc.

- **Inputs**
  - N is the graph size, and IsingType is the type of graph (for example: Spin Glass’s)
  - D0 is the eigenvalue dropout level
  - AlgIdler is the value of the idler signal in algorithm space.

- **Outputs**
  - T, Hmin are the Ising matrix and the ground state energy: $H_{\text{min}} = \min_j -\frac{1}{2} \sum_{i,j} T_{ij} \sigma_i \sigma_j$.
  - K, U, D0, J, D1, C are matrices used in the decomposition of the Ising problem. $K = U D_0 U^\dagger$ where K has been added a diagonal offset of amplitude depending on D0. $J = U \sqrt{D_0} U^\dagger$ and $C = 2J$.

- **Thresholds** is the threshold vector used in the algorithm (to define the nonlinear operation).

- **Inputs**
  - S, spins, In, H are initial reduced spins, spins, inputs states and energy.

- **Configure PNP**: Feeds the PNP with phases parameters, given an input and Ising problem. This function outputs a .txt file that can be read by the nanophotonic processor.

- **Inputs**
  - Input is the input in algorithm space (spins). It is the concatenation of a spin state of size N with the idler value in spin state (AlgIdler==1).

- **InputField** is the electric field input fed to the PNP, of the form $(E_0,0,0,0,0)^T$.

- **U** is the unitary matrix used in the diagonalization of the Ising problem.

- **h1, h2** are the homodyne detection matrices.
We detect output of the matrix multiplication in algorithm space. Also applies electronic feedback operations: diagonal multiplication, digital noise, nonlinear threshold, computes energy, etc.

**Output reader:** Given data from two homodyne detection, reconstructs output of the matrix multiplication in algorithm space. Also applies electronic feedback operations: diagonal multiplication, digital noise, nonlinear threshold, computes energy, etc.

**Inputs**
- `Out1`, `Out2` are the two detected output from the detector
- `D1`, `Thresholds`, `T`, `AlgIdler` are defined above.
- `DIGITAL_NOISE_FLAG` is a boolean defined whether extrinsic noise should be added to the output. `Phi` is the digital noise level.
- `ChipIndex` is the chip index ∈ (1, 2)

**Outputs**
- `NextFeed`, `NextS`, `Nextspins` are the next input, reduced spin and spin states.
- `NextE` is the new energy.

### A. Initialization and first feed-in

We first generate all relevant variables for a given problem with the General Initialization. The outputs of this function need to be global parameters as they will be useful throughout the run of the algorithm. Then, we use the current input `Input(:, 1)` and `ChipIndex=1` to generate phase parameters to code for the unitary matrices `h_1 U^T R_{In(:,1)}` and `h_2 U^T R_{In(:,1)}` where `h_1`, `h_2` are two homodyne detection matrices (`h_1` (resp. `h_2`) interferes outputs `i` and `i+1` starting with `i = 1` (resp. `i = 2`)), `R_{In(:,1)}` is a unitary matrix rotating `(1,0,...,0)^T` into `In(:,1)` and `U^T` is the unitary matrix present in the eigenvalue decomposition of our Ising weight matrix. Phases are generated assuming we can tune an `N x N` array of MZIs arranged in a rectangular pattern. This algorithm needs to be run twice on the PNP (corresponding to two homodyne detections):

1. Configure the PNP with phases `MZI_Array11`. We save the detected output as `Out11`
2. Configure the PNP with phases `MZI_Array12`. We save the detected output as `Out12`

### B. Output reading of PNP round 1, feed-in PNP round 2

We detect `Out11` and `Out12`, then to convert detected data into `Input(:, 2)`. Then, we configure the PNP with this new input `Input(:, 2)` and `ChipIndex=2`.

1. Calibrate PNP with phases `MZI_Array21`. We save the detected output as `Out21`
2. Calibrate PNP with phases `MZI_Array22`. We save the detected output as `Out22`

### C. Output reading of PNP round 2, feed-in PNP round 3

We detect `Out21` and `Out22`, then convert detected data into `Input(:, 3)` (data in algorithm space). Then, we configure the PNP with this new input `Input(:, 3)` and `ChipIndex=1`.

### D. Recurrent step

Every algorithm step translates into 4 runs on the PNP, corresponding to two unitary matrices, and two homodyne detections per unitary matrix.

- Round-trip on PNP 1, coding for `U^T`
  - Homodyne detection 1 : `h_1 U^T R_{In(:,2+ChipIndex+1)}`
  - Homodyne detection 2 : `h_2 U^T R_{In(:,2+ChipIndex+1)}`

- Round-trip on PNP 2, coding for `U`
  - Homodyne detection 1 : `h_1 U R_{In(:,2+ChipIndex+2)}`
  - Homodyne detection 2 : `h_2 U R_{In(:,2+ChipIndex+2)}`

### E. Homodyne detection algorithm

In the main text, we compare the experimental results to two kinds of simulation. The simulation refers to as “ideal phase-intensity reconstruction” is the recurrent transformation discussed in [2]. For the “homodyne phase-intensity reconstruction”, we also simulate the propagation of light on-chip through the MZI array. The two homodyne detection matrices we use are:

\[
 h_1 = \begin{bmatrix} s & s & 0 & 0 \\ -s & 0 & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & -s & 0 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & s & s & 0 \\ 0 & 0 & s & s \\ 0 & 0 & 0 & -s \end{bmatrix} \tag{S1}
\]

where `s = 1/\sqrt{2}`. In order to reconstruct the real output `x` from the two homodyne detection outputs, we apply the following cascaded algorithm.

#### Algorithm S.1. Cascaded homodyne algorithm for PNP output reconstruction.

Input data `x` to reconstruct

\[
 y_1 \leftarrow h_1 x \\
 y_2 \leftarrow h_2 x \\
 x_{Ref} \leftarrow 1 \\
 x_i \leftarrow y_2(i) \text{ (reconstructed x vector)}
\]

For all `i ∈ {2, ..., N}` do

- If `i` is even then
  \[
  d \leftarrow y_1(i) - y_1(i-1) \\
  x_i \leftarrow d x_{Ref}^{(i-1)}
  \]
- If `i` is odd then
  \[
  d \leftarrow y_2(i) - y_2(i-1) \\
  x_i \leftarrow d x_{Ref}^{(i-1)}
  \]

A potential issue with the above algorithm is the cascading of error throughout the reconstruction of the output. Also, if `x_i = 0`, then in theory `x_{i+1}` diverges. We suggest a more general class of algorithms in order to prevent these divergences, assuming we have access to more channels. Since we want to extract `2N` variables (the phase and amplitude of `N` PNP outputs), the reconstruction algorithm requires `2N` measurements. Let us assume we have access to `1 ≤ k ≤ N` local oscillators of known amplitude and phase, in order to interfere them with the `N` PNP outputs.

- If `k = 1`, the algorithm is similar to Algorithm S.1.
If $k = N$, the issue of algorithmic divergence is avoided. However, the effective footprint of the PNP is doubled compared to the one required to implement Algorithm S1.

Thus, by choosing $1 < k < N$, we can find a middle ground between optimizing the chip footprint and the issue of algorithmic divergences.

### F. Decomposition Algorithm

To perform an arbitrary unitary matrix multiplication on-chip, we need to decompose the matrix into a product of $2 \times 2$ unitary transformations, arranged in a lattice that corresponds to the shape of our MZI array. The unit block of our decomposition corresponds to a transformation through a single Mach-Zehnder Interferometer:

$$T_{n,n+1}(\theta, \phi) = \begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 1 \\ 1 & i \end{bmatrix}$$

Thus, we use the fact that for every unitary transformation $T$, there exists $D'$ such that $DT^{-1}(\theta', \phi') = T(\theta, \phi)'D'$, with the following conditions on the different phases:

$$\theta' = \theta$$
$$\phi' = \phi_1 - \phi_2$$
$$\phi'_1 = \phi_2 - \pi - \theta$$
$$\phi'_2 = \phi_2 - \pi - \theta$$

where $D = \begin{bmatrix} e^{i\phi_1} & 0 \\ 0 & e^{i\phi_2} \end{bmatrix}$ and $D' = \begin{bmatrix} e^{i\phi'_1} & 0 \\ 0 & e^{i\phi'_2} \end{bmatrix}$. We thus get the following decomposition in the end:

$$U = \left( \prod_{(m,n) \in S_L} T_{m,n} \right) \left( \prod_{(m,n) \in S_R} T^{-1}_{m,n} \right) D'$$

### 2. PNP Configuration and Calibration

#### A. Experimental setup configuration

The programmable nanophotonic processor (PNP) chip used in our experiments is a 26-mode silicon-on-insulator photonic integrated circuit fabricated by the OpSIS foundry. Coherent continuous-wave light at 1550nm, rotated into TE polarization, is coupled to the input of the PNP chip via a laser-written glass interposer fanout chip. These interposer chips, fabricated by TEEM Photonics, interface standard polarization-maintaining SMF telecommunication fiber (10µm mode field diameter) to the 2µm mode field diameter of the on-chip edge couplers and have characterized insertion losses in the range [0.47dB, 0.87dB] per channel. The optical signal vector is encoded in the amplitude and phase of guided light within the 26 on-chip waveguide modes. To accommodate space constraints and avoid global phase differences on the input, a single input port is excited and routed (shown in red, Fig. S1) to a 5x5 program unit (shown in green, Fig. S1). The program unit encodes a unitary matrix product of the state preparation, Ising unitary, and homodyne detection matrices, as detailed in Section I.I.

The resultant program unit outputs are routed to an array of InGaAs p-i-n photodiodes with responsivities of approximately 0.93 A/W for 1550nm and a 2GHz cutoff frequency. Intensity readout is limited, in practice, to approximately 4kHz by the rise time of the readout circuitry. The noise floor of the detectors is observed to be on the order of 2nW, with a saturation power of 20µW. In order to counteract errors resulting from the division step in the homodyne reconstruction algorithm (Section I.5), PNP outputs are only accepted if the total output power of the signal modes exceeds 1µW.

Coupling efficiency for the Gaussian waveguide beams to and from the chip decreases exponentially with the degree of misalignment between the fiber array and the chip facet. Once coupled, this misalignment can often occur due to the difference in the thermal expansion coefficients between the silicon chip and the glass interposer. To maintain stable coupling in the presence of thermal fluctuations from the environment and the heat-dissipating phase shifters on the chip, it is necessary to implement a thermal control system to stabilize the temperature on the PNP. We mount the chip and a thermistor on a copper block with thermally conductive paste. The thermistor measures the temperature of the chip and, in combination with a Peltier cooler glued to the copper block, a p-i-d feedback loop is established with an Arroyo Instruments TEC temperature controller to maintain the on-chip temperature to within 0.01 Kelvin.
Fig. S1. Schematic of PNP program configuration. A single input port is excited and routed to a 5x5 program unit that encodes a matrix product of the state preparation, Ising unitary, and homodyne detection matrices. Output intensities are measured by an array of InGaAs photodiodes.

B. PNP calibration

Nonidealities including imperfections in fabricated waveguides and inaccuracy in voltage control can severely degrade single-shot fidelity of the optical matrix multiplication unit (OMMU). Motivated by decomposition schemes mapping an arbitrary unitary operator to an array of Mach-Zehnder interferometer (MZI) phases [3, 4], we extract the phase-voltage relationship for each voltage-controlled thermal phase shifter on the chip using an MZI calibration protocol detailed in [5].

It is straightforward to measure an intensity fringe that depends on an MZI’s internal phase shifter, \( \theta \); however, we observe from the MZI transfer matrix shown in Equation (S4) with internal phase difference, \( \theta \), and external phase difference, \( \phi \), that the intensities detected at the two output modes of an MZI are independent of \( \phi \). Nonetheless, accurate setting of the external phase shifters is crucial to the implementation of unitary matrices on the PNP when there are multiple MZI layers in the decomposition [3, 4]. One can still observe an intensity signature from the external phase shifters by programming four adjacent MZIs in the configuration shown in Figure S2. The left and right MZIs are programmed as symmetric beam splitters that implement the Hadamard operation

\[
H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}
\]

while the top and bottom MZIs are identities, thereby generating a meta-MZI that behaves like a traditional MZI where the internal phase difference of the meta-MZI can be tuned by varying the external phase shifters of the left and bottom MZIs.

Given an individual meta-MZI, we seek to zero out the relative phase difference between the meta-MZI’s internal arms as a result of the left and bottom individual MZI external phase shifters. In order to do this, we sweep the individual MZI external phase shifters (shown as yellow blocks in Figure S3) until the given meta-MZI performs a “swap” operation. The “swap” configuration indicates that there is zero phase difference between the internal arms of the meta-MZI.

For each meta-MZI, there are two tunable phase shifters and two constant phase shifts inherent to the fabricated structure that, together, produce the net phase difference between the meta-MZI’s internal arms. In addition, adjacent meta-MZIs share one phase shifter. Relating the swap phase condition for neighboring meta-MZIs results in a system of \( m \) coupled equations in \( m \) variables, where each chain of meta-MZIs has one phase shifter (denoted in Figure S3 by the top yellow box outlined in purple) that serves as a reference phase, set to \( \pi \). We calculate the number of non-reference external phase shifters that we must calibrate to be given by

\[
m = (\# \text{ meta-MZIs}) - (\# \text{ meta-MZI "chains"}) = 63 - 11 = 52
\]

Furthermore, for each meta-MZI, the swap condition necessitates the following relationship between the tunable and intrinsic phases:

\[
\phi_{\text{top}} - \phi_{\text{bottom}} + \phi_{\text{config,diff}} + \Delta = 0 \quad (\text{S12})
\]

where \( \phi_{\text{top}} \) and \( \phi_{\text{bottom}} \) are the external phase shifters for the left and bottom individual MZIs, respectively, \( \phi_{\text{config,diff}} = \pi \) (implementation of the identity requires a phase difference of \( (\theta, \phi) = (\pi, \pi) \) for the top and bottom individual MZIs), and \( \Delta \) is the inherent phase difference between the two arms from variations in the waveguide fabrication.

The inherent phase difference, \( \Delta \), is fitted from the transmission sweeps of both \( \phi_{\text{top}} \) and \( \phi_{\text{bottom}} \) for a given meta-MZI. Using the fitted \( \Delta \) intrinsic phase difference for each meta-MZI,
we solve a simple matrix equation to determine the phase shifter offsets that cancel out the effect of fabrication imperfections on the external phase shifter. We also use these fitted transmission sweeps to extract a phase vs. voltage relationship for each individual external phase shifter.

C. PNP characterization

Given the calibrated phase vs. voltage functions for each of the individual phase shifter on the PNP, we seek to determine how accurately we can implement a desired unitary transform, with elements shown below:

\[
U = \begin{bmatrix}
    r_{11}e^{i\theta_{11}} & r_{12}e^{i\theta_{12}} & \cdots & r_{1n}e^{i\theta_{1n}} \\
    r_{21}e^{i\theta_{21}} & r_{22}e^{i\theta_{22}} & \cdots & r_{2n}e^{i\theta_{2n}} \\
    \vdots & \vdots & \ddots & \vdots \\
    r_{n1}e^{i\theta_{n1}} & r_{n2}e^{i\theta_{n2}} & \cdots & r_{nn}e^{i\theta_{nn}}
\end{bmatrix}
\] (S13)

We measure the magnitudes, \( r_{ij} \), and phases, \( \theta_{ij} \), using two different protocols (outlined in Figure S4).

Amplitude Measurement

\[ I_j = |r_{ij}|^2 \] (a)

Phase Measurement

\[ I_j(\phi) = |r_{ij}^2 + r_{1j}^2 + 2r_{1j}r_{ij}\cos(\phi + \theta_0)| \] (b)

We measure the magnitudes, \( r_{ij} \), and phases, \( \theta_{ij} \), on the output arms.

We measure the amplitude \( r_{ij} \) by measuring the output power on the \( j \)th detector \( I_j \) upon illumination of input \( i \) by with intensity \( I \):

\[ |r_{ij}| = \sqrt{\frac{I_j}{I}} \] (S14)

Measurement of \( \theta_{ij} \) is slightly more involved, as interference is required to observe an intensity signature from these phase terms. This interference is implemented using a protocol developed by [7] (outlined in Figure S4). We insert a coherent state input to the chip, program the MZI in the first layer to be a 50:50 beamsplitter, route the two split coherent states to input mode 1 and \( i \) of the unitary matrix circuit, and measure from output mode \( j \) while sweeping the value of a tunable phase difference, \( \phi \), between the split coherent states. Because the matrix is unitary and the columns and rows form a unitary basis, we can assume, without loss of generality, that the first column and first row of the characterized matrix are real [7]. We then use these “border” elements as references upon which to calculate the rest of the phases \( \theta_{ij} \). We first find the value of \( \phi \) that maximizes the intensity fringe generated at output \( j \) when we sweep \( \phi \). Fitting

the resultant fringe with the following relation subsequently gives us the value of \( \theta_{ij} \)

\[
I_j(\phi) = \left| r_{1j} + r_{ij}e^{i(\phi + \theta_0)} \right|^2 = |r_{1j}^2 + r_{ij}^2 + 2r_{1j}r_{ij}\cos(\phi + \theta_0)| \] (S15)

D. PNP voltage crosstalk correction

Programming an N-dimensional unitary matrix requires at least \((N)(N-1)\) phase shifters to be programmed simultaneously. When programming many MZIs at once, it is necessary to consider non-idealities in the structure of the resistive network driving the thermo-optic phase shifters in order to correct for voltage crosstalk in the electronic circuitry (shown in Figure S5a). In particular, the ground terminal comprises of a network of physical metal pours both on the PNP chip and on the control PCB, leading to a small resistance between the heater grounds and true ground. With each heater that is turned on, a small current flows through the ground resistor, changing the effective voltage drop across all the other resistors in the network.

![Fig. S4. Unitary matrix characterization protocol. (a) Magnitudes are measured using an intensity measurement at the detectors. (b) Phases are deduced from fitting the interference fringe at a selected output that results from two input modes being simultaneously illuminated, with a variable phase difference between the input arms.](image)

To characterize the effect of this ground resistance on the 240 pins of the driver+ board, we measure the 240-dimensional crosstalk coupling matrix (shown in Figure S5b). For each of the 240 voltage driver pins, a single pin was probed while each of the other 239 pins was swept in voltage between 0 Volts and 7.0 Volts in 10 increments. The resulting voltage crosstalk curves are not perfectly linear, however, so we apply a polynomial correction model described in the following paragraph, and outlined further in [8].

We observe increasingly nonlinear behavior in crosstalk curves at higher voltages, possibly due to the change in heater resistance due to decreased carrier mobility at higher temperatures [9]. We, therefore, model each measured voltage crosstalk curve as a cubic function of the set voltages, \( \bar{V}_{\text{set}} \):

\[
\bar{V}_{\text{meas}} = \bar{V}^{(1)}_{\text{set}} + \bar{V}^{(2)}_{\text{set}} + \bar{V}^{(3)}_{\text{set}}
\] (S17)

and use Newton’s method [10] to find a solution vector, \( \bar{V}_{\text{set}} \) for this nonlinear system of equations that produces a desired measured voltage vector.

To test the effect of this protocol, a sample of 100 unitary matrices were generated at random from using the following
procedure:

\[
C = \text{random complex } NxN \text{ matrix} \quad (S18)
\]

\[
H = \frac{1}{2}(C + C^\dagger) \quad (S19)
\]

\[
U = e^{iH} \quad (S20)
\]

After use of the characterization scheme outlined in Section II.3, we calculate the fidelity of the programmed matrices using the following unitary fidelity metric, with results shown in Figure S6.

\[
F = \frac{\text{Tr}(U^\dagger_t U)}{N}. \quad (S21)
\]

Fig. S6. Fidelities of 100 randomly generated U(5) unitary matrices with and without voltage crosstalk correction. Polynomial crosstalk correction assumes a cubic voltage crosstalk relation, solved using Newton’s method. Linear crosstalk correction assumes a linear relation, enabling a solution to be obtained by inverting the crosstalk correlation matrix.

3. PROPOSED LARGE-SCALE IMPLEMENTATION

In this work, we presented a proof-of-principle demonstration of a 4-spin system with the unitary matrix multiplication steps performed in an actively-controlled integrated photonic circuit. Although the recurrent transform is a static function of the Ising graph, we encoded the state preparation, recurrent update, and homodyne detection matrix product into one 5x5 unitary program that dynamically updates with each pass through the chip. A final version of this system will have separate circuit elements for each of the following components: state preparation, unitary matrix multiplication units (UMMU) for \(U^\dagger\) and \(U\), a non-unitary diagonal matrix multiplication unit (DMMU), nonlinear thresholding, and homodyne detection. The aim of such a system is to exploit the static nature of the matrix transform to perform each algorithm iteration very quickly and with low power consumption by using as many optical and passive components as possible. A diagram of the structure of a single algorithm step of the proposed system is shown in Fig. S7. Each component is described in more detail in the following paragraphs, and we show that the following system can be constructed to operate at GHz clock rates.

A. Spin state preparation

To generate the input spin state vector \(S^{(t)}\), one can feed a pulsed laser input to a passive binary splitting tree with \(\lceil\log(N)\rceil\) layers. Spins are encoded by Mach-Zehnder interferometer (MZI) switches set to either transmit full power or swap to a waveguide dump. Carrier depletion modulators that can operate at tens of GHz have been demonstrated in silicon photonics [12, 14], and any phase-dependent loss is not a concern in this application, as each phase shifter will impart only one of two phases: \(\{0, \pi\}\).

B. Recurrent Ising matrix program

The recurrent matrix update is a static function of the Ising graph and can be decomposed to a product of the following static matrices: \(K = UDU^\dagger\), where \(U\) and \(U^\dagger\) are unitary matrices and \(D\) is a diagonal matrix. The unitary matrix multiplication units could be fabricated with a chalcogenide-assisted silicon-on-insulator waveguide process [13, 15] in the MZI mesh structure presented in this work and optically trimmed in post-processing to encode the desired Ising unitary matrices. This trimming requires a constant overhead for each graph and enables the photonic matrix multiplication steps to be passive for all subsequent algorithm iterations. If fast updating of the dropout hyperparameter, \(\alpha\), is not required, the diagonal matrix can be similarly implemented as a single layer of MZI modulators and trimmed to the correct transmission using the same process. However, in general one would benefit being able to tune the dropout [2], in which case the diagonal matrix multiplication unit can be implemented using the same optical modulation techniques as those used for the state preparation circuit described above. For a system of 1000 spins and a sample MZI length of 40\(\mu m\), the propagation time through the two unitary matrix multiplication units and the diagonal matrix multiplication unit is approximately 0.5\(ns\).

C. Nonlinear threshold

Following the matrix multiplication units, a nonlinear threshold function acts on the output fields to generate the spin input for the subsequent iteration. This can be done either optically through components such as graphene-on-silicon saturable absorbers [16], optoelectronically using a ring resonator modulated with a detection feedback loop, [17] or by performing homodyne detection on the output using a local oscillator fan-out and the nonlinear processing as a single step in electronics. Each of these implementations is limited by the bandwidth of the detection electronics, which for InGaAs photodiode circuits have been demonstrated with operation at \(\sim 10\text{GHz}\) [18].

4. BENCHMARKING

A. Time Scaling

In this section, we compare the following figures of merit for our architecture, several other state-of-the-art physical Ising machines, and the D-Wave quantum annealer: number of algorithm iterations necessary to output the correct ground state with 90\% probability \(N_{90\%}\) of a dense MAXCUT graph, time per algorithm iteration \(T_{iter}\), and subsequent time to obtain 90\% ground state probability \(T_{90\%}\). The solution times for the D-Wave 200Q machine [19] are standardized using the following definitions, where \(T_{ann}\) is the time of a single annealing run and \(p\) is the true probability of finding the ground state.
Fig. S7. (a) System diagram of the components of a single iteration step in a proposed large-scale implementation. (b) The desired Ising graph problem is processed and encoded onto a photonic circuit program consisting of Mach-Zehnder interferometers (MZI). This program consists of a decomposition of a function of the Ising graph matrix into a product of two unitary matrix multiplication units (UMMU) separated by a diagonal matrix multiplication unit (DMMU) (c) A single MZI consists of two phase shifters that can be implemented using thermo-optic phase shifters[11], carrier-depletion phase shifters [12], or chalcogenide-assisted trimmed waveguides [13]. The final approach enables quasi-passive operation and high operating speeds with lower power consumption than the active phase shifter techniques. For a given algorithm step, an input spin state $S(t)$ propagates through the UMMUs, DMMU, and a nonlinear threshold element to project the continuous field outputs back into spin space. (d) The output of the iteration, $S(t+1)$, is a spin state that is sampled from a Gibbs distribution of the Ising Hamiltonian.

It is important to note that the iteration structure differs between the INPRIS presented in this paper and the optical parametric oscillator (OPO) and D-Wave annealers. The INPRIS can sample a new point in the phase space with every iteration, therefore, the iteration time $T_{iter}$ is defined as a single round trip through the system. However, the OPO Ising machine and D-Wave converge to a single spin state with each annealing run and must be restarted to converge to a different state. Therefore, the iteration time for this comparison is defined as a single annealing step, $T_{ann} = \frac{NR}{f}$, where $N$ is the graph order, $R$ is the number of round trips through the system, and $f$ is the laser pulse repetition rate.

The results for the INPRIS in Table S1 are listed for a machine with 16-bit voltage precision. Although the digital to analog converters in our control electronics enable 16-bit precision, voltage crosstalk degrades this precision. An estimate for the realistic precision is calculated below, given an estimated crosstalk per phase shifter of 20 $\mu$V/V (From Fig. S5(b)). We correct for this reduction in bit-depth by implementing the voltage crosstalk correction protocol described in SI, Section 2D. With correction at every algorithm step, the precision of the system does not prevent successful ground state search. Voltage crosstalk can be avoided in future implementations by using current drivers and reducing the number of active photonic circuit components.

\[
\Delta V_{\text{talk}} \approx (V_x)(20\mu V / V)(N)(N-1) \tag{S24}
\]
\[
= (4.36V)(20\mu V / V)(5)(4) = 1.744mV \tag{S25}
\]
\[
\frac{8.72V}{2^b} = 1.744mV \tag{S26}
\]
\[
\implies [b] = 12 \text{ bits}. \tag{S27}
\]

<table>
<thead>
<tr>
<th>Architecture</th>
<th>$N_{90%}$</th>
<th>$T_{iter}$ ($T_{ann}$)</th>
<th>$T_{90%}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPRIS, 16-bit [2]</td>
<td>$10^4$</td>
<td>0.1-1ns</td>
<td>1-10$\mu$s</td>
</tr>
<tr>
<td>PRIS, FPGA [2]</td>
<td>$10^4$</td>
<td>5-100ns</td>
<td>50$\mu$s-1ms</td>
</tr>
<tr>
<td>OPO CIM [19, 20]</td>
<td>$10^3$</td>
<td>(250ns)</td>
<td>1ms</td>
</tr>
<tr>
<td>SLM CIM [21]</td>
<td>N/A</td>
<td>1ms</td>
<td>N/A</td>
</tr>
<tr>
<td>D-Wave 2Q* [19]</td>
<td>$10^{21}$</td>
<td>(1ms)</td>
<td>$5 \times 10^{18}$ s</td>
</tr>
</tbody>
</table>

Table S1. Estimated time scaling figures for the proposed experimental architecture and other competing digital and physical Ising machines. We consider dense MAX-CUT graphs of order $N=100$ for each of the machines.

B. Energy consumption

We can calculate the approximate power consumption of a proposed quasi-passive system as a sum of the power consumption from the following algorithm components (delineated in SI, Section 3): pulsed laser power, spin state preparation, optical diagonal matrix multiplication unit, nonlinearity, and homodyne detection (detectors, circuitry, and a local oscillator).
\[ P \approx N(P_{\text{state prep.}} + P_{\text{DMMU}} + P_{\text{nonlinearity}} + P_{\text{det}} + P_{\text{TIA}} + P_{\text{LO}}). \]  

The ring modulators presented in [22] exhibit a switching power of approximately 20 \( \mu \)W. We assume use of these modulators for the state preparation and DMMU components. For a waveguide geometry of 220nm \( \times \) 500nm, the threshold requirement of many typical saturable absorber optical nonlinearities is approximately 1mW of optical power [23]. Typical InGaAs photodiodes, such as those used in the presented experimental demonstration, can have tens of GHz of bandwidth, a responsivity of approximately 1A/W, and a drive voltage of approximately 5V, leading to a power consumption of 5mW for each detector with 1mW of incident optical power. Given homodyne detection requires two detectors, we set \( P_{\text{det}} = 10N \) mW. Low-power, high bandwidth trans-impedance amplifiers with power consumption of approximately 2mW have been demonstrated in [24]. Finally, a strong local oscillator signal is required for interfering with each circuit output. We estimate a figure of 5mW per local oscillator field. Using all these figures, we obtain a power estimate of

\[ P_{\text{system}} \approx N(20\mu W + 20\mu W + 1mW + 10mW + 2mW + 5mW) = (18N)mW \]

5. EXTENDED RESULTS ON GROUND STATE SEARCH

The dependence of the ground state population as a function of the extrinsic noise is plotted for all studied graphs and two levels of dropout (\( \alpha = 0 \) and \( \alpha = 1 \)) in Figure S8. Additionally, the measured spin state distribution for all studied graphs and dropout levels are compared to the Gibbs distribution of the corresponding Ising Hamiltonian and a random search of the phase space in Figure S9.

6. PHASE SPACE EXPLORATION CHARACTERIZATION

In this section, we represent the Ising phase space for a variety of graphs and extrinsic noise level \( \phi = 1.0 \). In each plot in Figures S10, S11, S12, and S13, the \( y \) (resp. \( x \)) coordinate corresponds to the binary representation of the spins \( (\sigma_1, \sigma_2) \in \{(-1, -1), (-1, +1), (+1, -1), (+1, +1)\} \) (resp. \( (\sigma_3, \sigma_4) \)). The area of each dot is proportional to the probability of spin state \( (y, x) \), which is also encoded in the color of the dot. At each coordinate, three concentric dots are plotted, with respective areas proportional to the mean probability, the mean probability minus its standard deviation, and the mean probability plus its standard deviation (all statistics for each plot are averaged over 10 runs of the same graph with random initial states).
Fig. S8. Extended results on ground state search. Population plots and histograms are shown for antiferromagnet (A, B), ferromagnet, (C, D), random cubic graph (E, F), random sparse graph (G, H), spin glass B (I, J), and spin glass A (K, L). (A, C, E, G, I) correspond to dropout ($\alpha = 0$) and (B, D, F, H, J, L) to no dropout ($\alpha = 1$).
Fig. S9. Ground state convergence to Gibbs distribution. Histograms of measured output spin state energy are shown for the extrinsic noise level, $\phi_{\text{opt}}$, that optimizes ground state probability. Histograms for the Gibbs distribution using each Ising graph Hamiltonian are shown in red, and the corresponding histogram values for a random search are shown in grey.
Fig. S10. Phase space representation of antiferromagnetic and ferromagnetic graphs with PNP (left), simulation with homodyne detection (center), and simulation without homodyne detection (right).
Fig. S11. Phase space representation of random cubic and random sparse graphs with PNP (left), simulation with homodyne detection (center), and simulation without homodyne detection (right).
Fig. S12. Phase space representation of two spin glasses with PNP (left), simulation with homodyne detection (center), and simulation without homodyne detection (right).
Fig. S13. Phase space exploration for ferromagnetic graph at various noise levels.
REFERENCES