Supplementary Information for: "Recasting maximum-likelihood sequence estimation as optoelectronic Ising dynamics for bandwidth-limited optical interconnects"

***Yibin Wan*,[1] *Zhenhua Li*[1], *Zihao Chen*[1], *Zhixian Zhou*[1], *Junyi Liu*[1], *Jiayi Gao*[1], *Jiakai Dong*[1], *Jiaqing Li*[1], *Jie Liu*,[1,*] *Siyuan Yu*[1]**

[1] State Key Laboratory of Optoelectronic Materials and Technologies, School of Electronics and Information Technology, Sun Yat-Sen University, Guangzhou 510006, China

*Corresponding author: liujie47@mail.sysu.edu.cn.

Supplementary Note 1: Recasting MLSE as Ising energy minimization

The core of recasting MLSE as Ising energy minimization lies in establishing an equivalence between the cost function of maximum likelihood sequence detection (MLSE) and the Hamiltonian of the Ising model. This involves three essential steps: first, constructing a mapping from the noise-free symbols—originally discrete but not necessarily binary—to binary spin variables that are native to the Ising formulation; second, simplifying the MLSE problem without affecting its solution, making it similar in mathematical form to the Ising model; third, deriving the corresponding coupling fields and external fields that characterize the resulting Ising Hamiltonian. Once these parameters are obtained, the MLSE problem is reformulated as an energy minimization task, allowing the Ising machine to efficiently seek the optimal or near-optimal transmitted sequence through its natural dynamical evolution.

The cost function of MLSE can be expressed as:

$$\begin{aligned}\hat{\boldsymbol{x}}_{\boldsymbol{ML}} &= \arg\min_{\boldsymbol{x}\in\Omega^{K}} \left\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\right\|^{2} \\ &= \arg\min_{\boldsymbol{x}\in\Omega^{K}} \boldsymbol{x}^{T}\boldsymbol{H}^{T}\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{y}^{T}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{y}^{T}\boldsymbol{y}\end{aligned} \tag{S1}$$

where Ω is the alphabet of all possible modulation symbols and K is the sequence length. MLSE estimates the transmitted symbol sequence $\boldsymbol{x}$ by minimizing the Euclidean distance between the received sequence $\boldsymbol{y}$ and the channel output predicted by an equivalent channel matrix $\boldsymbol{H}$. The channel modeling in this work can be found in Supplementary Note 2.

To express this search in Ising form, each transmitted symbol is encoded by one or more binary spins. Assuming the use of PAM-M modulation with $M = 2^{Z}$, the discrete symbol, x_i, can be modeled using spin variables as:

$$x_i = \sum_{j=1}^{Z} 2^{Z-j}\sigma_{i+(j-1)K} \tag{S2}$$

For example, for PAM-2, $Z = 1$, $\Omega = \{\pm 1\}$, thus $x_i = \sigma_i$; for PAM-4, $Z = 2$, $\Omega = \{\pm 1, \pm 3\}$, thus $x_i = 2\sigma_i + \sigma_{i+K}$. Let us define a transformation matrix $\boldsymbol{T} = \boldsymbol{I}_K \otimes [2^{Z-1}, 2^{Z-2}, \cdots 1]$, $\boldsymbol{I}_K$ is $K\times K$ unit matrix. Thus, we can obtain the symbol-to-spin mapping in vector form:

$$\boldsymbol{x} = \boldsymbol{T}\boldsymbol{\sigma}\ ,\ \boldsymbol{\sigma} \in \{-1, +1\}^{KZ} \tag{S3}$$

Substituting $\boldsymbol{x} = \boldsymbol{T}\boldsymbol{\sigma}$ into the MLSE metric gives a quadratic spin objective,

$$\left\| \boldsymbol{y} - \boldsymbol{H}\boldsymbol{T}\boldsymbol{\sigma} \right\|^2 = \boldsymbol{\sigma}^T \boldsymbol{Q}\boldsymbol{\sigma} - 2\boldsymbol{b}^T\boldsymbol{\sigma} + \boldsymbol{y}^T\boldsymbol{y} \tag{S4}$$

with

$$\boldsymbol{Q} = \boldsymbol{T}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{T},\ \boldsymbol{b} = \boldsymbol{T}^T\boldsymbol{H}^T\boldsymbol{y} \tag{S5}$$

The constant term, including $\boldsymbol{y}^T\boldsymbol{y}$ and the diagonal contribution of $\boldsymbol{Q}$, does not depend on the spin configuration and therefore does not affect the minimization. Removing this constant offset gives the standard Ising form:

$$\boldsymbol{\sigma} = \underset{\boldsymbol{\sigma}\in\{-1,+1\}^{KZ}}{\arg\min} - \boldsymbol{\sigma}^T\boldsymbol{J}\boldsymbol{\sigma} - \boldsymbol{g}^T\boldsymbol{\sigma} \tag{S6}$$

where

$$\boldsymbol{J} = -\,\mathrm{zeroDiag}(\boldsymbol{T}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{T}),\ \boldsymbol{g} = 2\boldsymbol{T}^T\boldsymbol{H}^T\boldsymbol{y} \tag{S7}$$

Thus, the channel response determines the spin coupling matrix $\boldsymbol{J}$, while the received waveform determines the external field $\boldsymbol{g}$ through projection by $\boldsymbol{H}^T$ .This establishes the direct equivalence between MLSE and Ising energy minimization. The Ising parameter $\boldsymbol{J}$ and $\boldsymbol{g}$ are extracted from the communication-defined MLSE problem and subsequently the Ising-MLSE problem is solved by the Ising machine. Finally, the resulting spin configuration can be mapped back to the estimated transmitted sequence through $\boldsymbol{x} = \boldsymbol{T}\boldsymbol{\sigma}$.

Supplementary Note 2: Channel modeling based on noise whitening filter coefficients

It is well established from the Nyquist sampling theorem that the maximum symbol rate free of ISI is twice the system bandwidth. In our bandwidth-limited IM-DD optical interconnect system, which has an analogue bandwidth of only 55 GHz and operates beyond 120 Gbaud, the inevitable ISI must be effectively mitigated at the receiver to ensure reliable error performance. At the receiver, linear equalizers such as FFE aim to minimize ISI, but it applies the same filtering to both signal and noise. This process colors the noise, thereby violating the optimality condition for symbol-by-symbol detection[1]. For MLSE detection under colored noise, a noise-whitening filter (NWF) is commonly applied before sequence detection. While whitening the noise, the NWF introduces controlled ISI to the signal. Based on the response coefficients of the whitening filter, an effective channel model can be constructed for MLSE,

which eliminates the introduced ISI and achieves optimal signal detection. The entire process is described in detail below.

Suppose the discrete signal after FFE is expressed as:

$$s_t = x_t + \tilde{n}_t \tag{S8}$$

where x_t denotes the ideal noise-free symbol, and $\tilde{n}_t$ is the colored noise introduced by the equalization process. After applying a NWF, the discrete whitened signal in the presence of ISI is expressed as：

$$y_t = \sum_{i=1}^{L} h_i s_{t-i} = \sum_{i=1}^{L} h_i x_{t-i} + n_t \tag{S9}$$

where h_i is the tap coefficient of NWF, L is the channel response length, n_t is the whitened noise caused by NWF. This whitening process, while beneficial for noise statistics, reintroduces controlled ISI into the signal. A longer memory length can achieve higher degree of whitening at the cost of more severe ISI.

To derive the response coefficients of NWF, we take the autoregressive (AR) spectral estimate of the power spectral density (PSD) of the noise after FFE[2]. The noise $\tilde{n}_t$ is first estimated by subtracting the information signal d_t from the FFE output s_t, which can be expressed as:

$$\tilde{n}_t = s_t - d_t \tag{S10}$$

where the information signal d_t can be either training sequence or the direct decision of s_t. This is achieved by training sequences in our work. The whitening filter response coefficients are derived from the coefficients of the AR model of the noise. Here, Yule-Walker equations are used to extract the AR coefficients[3]. Since all the values are real numbers, the whitening filter response coefficients can be obtained as follows:

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{p+1} \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & \cdots & R(p) \\ R(1) & R(0) & \cdots & R(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ R(p) & R(p-1) & \cdots & R(0) \end{bmatrix}^{-1} \begin{bmatrix} R(0) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{S11}$$

where $R(i)$ is the autocorrelation of the noise in Equation S10. Assuming that a L-tap FIR filter is used for whitening the noise, we can set $p = L$-1 in Equation S11 to obtain the NWF response coefficients $\boldsymbol{h} = [h_1, h_2, \cdots, h_L]$.

Denote $\boldsymbol{y} = [y_L, y_{L+1}, \cdots, y_K]^T$ is a $(K-L+1)\times 1$ signal vector with ISI introduced by NWF, then we can further express Equation S9 in vector form as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{S12}$$

where $\boldsymbol{x} = [x_1, x_2, \cdots, x_K]^T$ and $\boldsymbol{n} = [n_L, n_{L+1}, \cdots, n_K]^T$ are a $K\times 1$ noise-free symbol vector and a $(K-L+1)\times 1$ noise vector, respectively. The Toeplitz $\boldsymbol{H}$ is a $(K-L+1)\times K$ equivalent channel matrix based on the response coefficients of NWF, which is elaborated below:

$$\boldsymbol{H}=\begin{bmatrix} h_L & h_{L-1} & \cdots & h_1 & & \\ & h_L & h_{L-1} & \cdots & h_1 & \\ & & \ddots & \ddots & \ddots & \ddots \\ & & & h_L & h_{L-1} & \cdots & h_1 \end{bmatrix} \tag{S13}$$

This equivalent channel matrix based on response coefficients of NWF provides an effective channel model in Equation S12, which can be used for the subsequent MLSE and Ising-MLSE to enable optimal sequence detection. The cost function of MLSE and recasting MLSE as Ising energy minimization can be found in Supplementary Note 1.

Supplementary Note 3: Dynamical evolution process of the optoelectronic Ising machine

Ising-MLSE is implemented using an optoelectronic Ising machine in our work, which follows an iterative computation procedure that includes nonlinear transformation and linear matrix-vector multiplication (MVM). The optoelectronic Ising machine primarily consists of simple optoelectronic components such as Mach-Zehnder modulator (MZM), photodetector (PD), and other processing devices, e.g., field-programmable gate array (FPGA) paired with analog-to–digital converter (ADC) and digital-to-analog converter (DAC)[4–7]. First, a Mach-Zehnder modulator (MZM) is used to perform electro-optic modulation on the optical signals emitted by a laser. In this process, the amplitudes of spin nodes are encoded into optical intensities in a time-multiplexing scheme; then the node amplitudes are converted into electrical signals via a PD. The process of MZM-based intensity modulation and PD detection inherently provides in-line nonlinear transformation operation capability, which avoids the additional optoelectronic energy consumption and latency overhead required for digital-domain implementation or realization via optical nonlinear effects[4].

The time evolution of the i-th spin amplitude $a_i^{(k+1)}$ during iteration step k+1 is given by the following nonlinear transfer function：

$$a_i^{(k+1)}=\cos^2\left(u_i^{(k)}-\frac{\pi}{4}+\eta_i^{(k)}\right)-\frac{1}{2} \tag{S14}$$

where $u_i^{(k)}$ is a feedback term of spin node, $\eta_i^{(k)}$ represents the system noise in optical pathway.

The resulting photovoltages after PD are sampled by an ADC, and the feedback values of spin nodes are obtained by loading self-feedback, spin coupling and external-field guidance according to the following formula：

$$u_i^{(k)}=\alpha a_i^{(k)}+\beta_J\sum_j J_{ij}a_j^{(k)}+\beta_g g_i+\xi_i^{(k)} \tag{S15}$$

Its corresponding vector-matrix form is given by:

$$\boldsymbol{u}^{(k)}=\alpha\boldsymbol{a}^{(k)}+\beta_J\boldsymbol{J}\boldsymbol{a}^{(k)}+\beta_g\boldsymbol{g}+\boldsymbol{\xi}^{(k)} \tag{S16}$$

where $\boldsymbol{\xi}^{(k)}$ represents the system noise in electrical pathway; α, β_J and β_g control the strengths of self-feedback, spin coupling and external-field guidance, respectively, where β_J and β_g can take the same value. This computation is essentially a linear MVM operation. In our work, it is performed by an FPGA (Xilinx KU115), paired with ADCs/DACs featuring a sampling rate of 2.6 GHz. Subsequently, the feedback signal is injected from a DAC to the MZM and remodulated into optical signal for next iterative computation step.

Supplementary Note 4: Sparse matrix-vector multiplication algorithm based on FPGA

Generally, a fully connected Ising graph needs a full-size MVM with high computational complexity in dynamical evolution process, leading to relatively large computation latency and energy consumption. Considering that the coupling matrix $\boldsymbol{J}$ containing only the temporal interactions induced by ISI is sparse, we have embedded an efficient sparse matrix–vector multiplication (SpMV) algorithm[5,6,8] into the FPGA to enable minimized computational complexity, facilitating Ising problem solving with low latency and energy consumption.

In this approach, the sparse matrix $\boldsymbol{J}$ needs to be converted into the Compressed Sparse Row (CSR) format. As in the example shown in Figure S1(a), the sparsely distributed non-zero elements in each row of matrix $\boldsymbol{J}$ are first densely stored sequentially into the vector *val*, while their corresponding column indices in $\boldsymbol{J}$ are recorded in the vector *col*. Then, the index of the first non-zero element of each row within the vector *val* is stored in the vector *ptr*. Elements belonging to the same row in vectors *val* and *col* can be identified according to *ptr*.

Subsequently, the CSR-format vectors *val* and *col* are loaded into the access-and-multiplication units (AMUs) in multiplication-accumulation core (MACC) of FPGA to perform SpMV operations. The architecture of each AMU is illustrated in Figure S1(b). Particularly, an element a_{col} of vector $\boldsymbol{a}$ (referenced in Equation S16) is accessed from the block random-access memory based on the index in the *col* and then multiplied with the corresponding matrix element *val* of *J* by a multiplier. Then, all the output results from AMUs in a MACC are accumulated by a summarization tree module to obtain a target element of vector $\boldsymbol{u}$ (referenced in Equation S16), as illustrated in Figure S1(c).

It can be seen that, compared with the theoretically required N^2 element-wise computational complexity for *N*-spin MVM, the proposed SpMV algorithm significantly reduces computational complexity by skipping all operations involving zero-valued matrix elements. Thereby, computing resources are maximally allocated to calculate the products of non-zero elements, reducing computational latency and energy consumption. Meanwhile, multiple MACCs deployed on the FPGA can simultaneously obtain multiple elements of vector $\boldsymbol{u}$ in parallel, which further improves the throughput of parallel computing and lowers the computational latency.

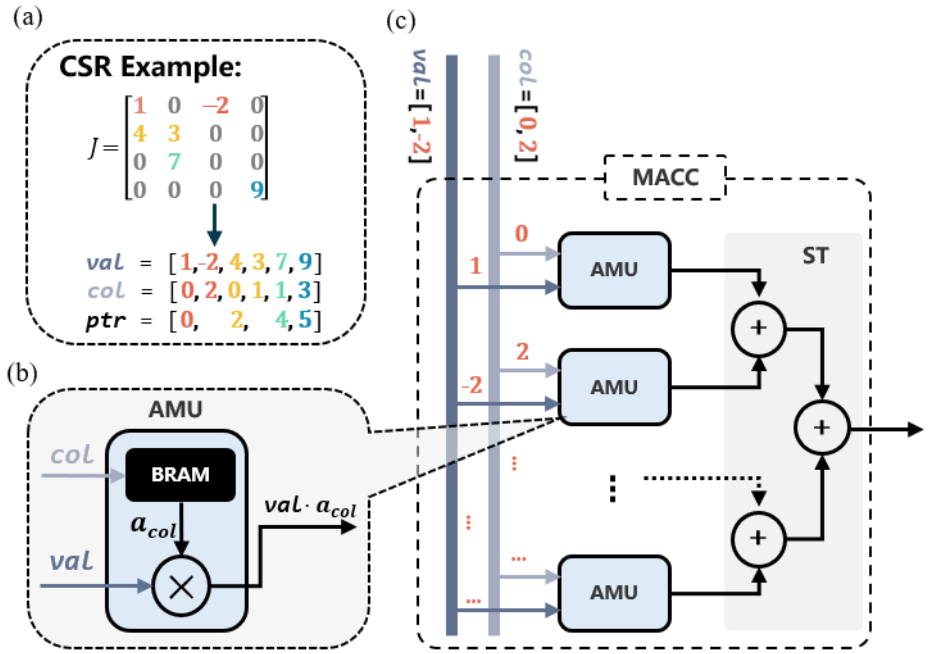

Figure S1: (a) Schematic diagram of CSR format conversion for a sparse matrix; (b) Internal architecture of the AMU; (c) Architecture of the SpMV MACC.

Supplementary Note 5: Computational complexity, energy and latency evaluation

In order to comprehensively evaluate the performance, we evaluated the computational complexity of brute-force, Viterbi-based, and Ising-based implementations of MLSE based on the number of multiplication and accumulation (MAC) operations in the computation. On this basis, we further derived the computational energy consumption and latency in the specific experimental implementations and predicted schemes.

Brute-force digital MLSE. For a brute-force implementation of digital MLSE, the receiver exhaustively evaluates all possible transmitted symbol sequences based on Equation S1. Given a modulation order M from the PAM-M constellation, a channel impulse response length L (i.e., the number of taps in the equivalent discrete-time filter), and a received sequence of length N, the total number of candidate symbol sequences is M^N. For each candidate sequence, the squared Euclidean distance metric is computed over N symbol intervals. At each interval, the reconstruction of the received signal requires L+1 multiplications to weight the symbols by the taps and L+1 additions to sum the weighted contributions, i.e., L+1 MAC operations. Hence, the metric calculation for one candidate sequence involves $(L+1)\times N$ MAC operations. The total number of MAC operations for the brute-force digital MLSE implementation is therefore:

$$\mathrm{MAC}_{\text{Brute-force}} = (L+1)\times N \times M^N \tag{S17}$$

This complexity grows exponentially with the sequence length N. Taking PAM-2 modulation with a channel response length $L = 8$ as an example, the computational complexity as a function of the sequence length is shown in Fig. 3a of the main text. Clearly, brute-force MLSE is

feasible only for very short sequences when we considering the resulting energy consumption and latency costs of practical implementation.

Viterbi-based digital MLSE. Digital MLSE performs signal detection by searching for the most probable sequence within the state trellis using the Viterbi algorithm[9]. When the channel impulse response has a length of L, the state of a sequence at any time instant is determined by its $L-1$ most recent symbols. As each symbol within a state takes one of M possible values from the PAM-M constellation, the trellis contains M^{L-1} states. Since each state has M outgoing transitions (or branches) in the trellis diagram, the total number of transitions within the state trellis is M^L. The computational complexity of the Viterbi algorithm is primarily governed by the calculation of transition metrics, which are further accumulated to update the path metric for each state. The computation of path metric for each state between adjacent time instants can be expressed as:

$$TM_t = \left| s_t - \sum_{k=0}^{L-1} h_k x_{t-k} \right|^2 \tag{S18}$$

$$PM_t = PM_{t-1} + TM_t \tag{S19}$$

where the TM_t evaluates the instantaneous transition cost, while the PM_t accumulates transition metrics. The calculation of each transition metric requires $L+1$ multiplications and L additions, and one additional addition is needed for path-metric accumulation. Therefore, the update of each path metric involves $L+1$ multiplications and $L+1$ additions, i.e., $L+1$ MAC operations per transition. Therefore, Viterbi-based digital MLSE requires $(L+1)\times M^L$ MAC operations for per symbol detection. For a received sequence of length N, the total number of MAC operations for the Viterbi-based digital MLSE implementation is therefore:

$$\mathrm{MAC}_{\mathrm{Viterbi}} = (L+1)\times M^L \times N \tag{S20}$$

Obviously, this computational complexity is linear in the sequence length N, which is a significant simplification compared to brute-force computation. Taking PAM-2 modulation with a channel response length $L = 8$ as an example, the computational complexity as a function of the sequence length is shown in Fig. 3a of the main text. However, it still grows exponentially with the channel response length (see Fig. 3b of the main text), which inevitably leads to an exponential increase in both energy consumption and latency in practical implementation.

In our works, the Viterbi-based digital MLSE is implemented on a digital computer (Intel Core i7-8700). The computational latency is directly measured through the MATLAB software. The average computational latency per symbol is obtained as follows. For each configuration (PAM-2 with $L = 7$ to 11, and PAM-4 with $L = 3$ to 7), we run the Viterbi-based MLSE 100 times. The total execution time is recorded, and the average execution time per MLSE run is computed. This average execution time is then divided by the number of symbols contained in one MLSE detection to yield the average latency per symbol. The resulting latency results for

PAM-2 and PAM-4 under different channel response lengths are presented in Fig. 6 of the main text.

For the energy consumption, the power P and clock rate B of the CPU are 65W and 300GOPS[10], respectively. By extensive observations, the CPU work accounts for approximately 30% when solving the Viterbi-based digital MLSE problem. Therefore, the energy efficiency of digital computing can be estimated as $\mu_{dig} = \frac{30\% \times P}{B} = 130\,\text{pJ/MAC}$. Then, the energy consumption for per symbol detection in our experiments can be expressed as

$$P_{dig} = \mu_{dig} \times M^{L} \times (L+1) \tag{S21}$$

Based on this evaluation method, the energy consumption results of PAM-2 and PAM-4 based on different channel response lengths are presented in Fig. 6 of the main text. Specifically, the energy consumption results for PAM-2 with a channel response length of 9 and PAM-4 with a channel response length of 5 are directly listed in Table 1. Additionally, we include a comparison with a general-purpose high-performance DSP core, the TI C66x, which achieves an energy efficiency of approximately 27 pJ/MAC in a 40-nm CMOS technology[10]. Its corresponding energy consumption under the same conditions is also presented in Table 1 as the comparison in predicted schemes.

Hardware-solved Ising-MLSE. For hardware-solved Ising-MLSE, the FPGA entails significant computational complexity in terms of MAC operations, as shown in Supplementary Note 3. For a channel response has a length of L, it implies that each symbol interacts with its L-1 neighboring symbols. After mapping the symbols to spins based on Equation S2, this results in each spin being coupled to $Z \times (L\text{-}1)$ other spins in coupling fields $\boldsymbol{J}$, where $Z = \log_2(M)$. Considering that we adopt a sparse MVM algorithm (see in Supplementary Note 4), the numbers of MAC required to solve each spin is $Z \times (L\text{-}1)$. Suppose the number of iterations is K and the number of runs is R, the total numbers of MAC for per symbol detection are $Z^2 \times R \times K \times (L\text{-}1)$. For a received sequence of length N, the total number of MAC operations for the hardware-solved Ising-MLSE implementation is therefore:

$$\text{MAC}_{\text{Ising}} = Z^2 \times R \times K \times (L-1) \times N \tag{S22}$$

This computational complexity is also linear in the sequence length N. Taking PAM-2 modulation with a channel response length $L = 8$ as an example, the computational complexity as a function of the sequence length is shown in Fig. 3a of the main text. The coefficients R and K are two adjustable parameters in the FPGA. In our experiments, they are fixed to 5 and 50, respectively. Furthermore, the computational complexity of the hardware-solved Ising-MLSE grows linearly with the channel response length, in contrast to the Viterbi-based digital MLSE, which exhibits exponential growth. The complexity comparison between the two schemes is shown in Fig. 3b of the main text.

In our experiments, the computational latency is measured through the protocol of communication between the FPGA and ADC/DAC[5]. For comparison with Viterbi-based digital

MLSE, the computational latency results for each configuration (PAM-2 with L = 7 to 11, and PAM-4 with L = 3 to 7) based on hardware-solved Ising-MLSE also are presented in Fig. 6 of the main text. It is clear that the hardware-solved Ising-MLSE is a more cost-friendly calculation method compared with Viterbi-based digital MLSE.

For the energy consumption, our FPGA achieves an energy efficiency of $\mu_{Ising} = 51.9\,\mathrm{pJ/MAC}$, detailed calculations were demonstrated in our previous work[5]. Therefore, the energy consumption for per symbol detection in our experiments can be expressed as:

$$P_{Ising} = \mu_{Ising} \times Z^2 \times R \times K \times (L-1) \tag{S23}$$

Based on this evaluation method, the energy consumption results compared with Viterbi-based digital MLSE are presented in Fig. 6 of the main text. Specifically, the energy consumption results for PAM-2 with a channel response length of 9 and PAM-4 with a channel response length of 5 are also directly listed in Table 1. For comparison with a more advanced hardware implementation, we also include a predicted optoelectronic Ising-machine scheme proposed in our previous work[5]. In that scheme, the multiplication operations required for matrix–vector multiplication are transferred from the FPGA to the on-chip thin-film lithium niobate modulator, so that the modulator concurrently performs electro-optic nonlinear transformation and element-wise multiplication, while the digital circuitry mainly completes the accumulation operations. With high-bandwidth, low half-wave voltage on-chip modulators, this architecture is expected to improve the energy efficiency to 2.4 pJ/MAC. The corresponding predicted energy consumption is also included in Table 1 for comparison.

References for Supplementary Note

1. Che, D. & Chen, X. Higher-Order Modulation vs Faster-Than-Nyquist PAM-4 for Datacenter IM-DD Optics: An AIR Comparison Under Practical Bandwidth Limits. *J. Light. Technol.* **40**, 3347–3357 (2022).
2. Liu, L. Li, L. & Lu, Y. Detection of 56GBaud PDM-QPSK generated by commercial CMOS DAC with 11GHz analog bandwidth. in *The European Conference on Optical Communication (ECOC)* 1–3 (2014).
3. Kay, S. M. & Marple, S. L. Spectrum analysis—A modern perspective. *Proc. IEEE* **69**, 1380–1419 (1981).
4. Böhm, F., Verschaffelt, G. & Van Der Sande, G. A poor man's coherent Ising machine based on opto-electronic feedback systems for solving optimization problems. *Nat. Commun.* **10**, 3538 (2019).
5. Li, Z. *et al.* Scalable On-Chip Optoelectronic Ising Machine Utilizing Thin-Film Lithium Niobate Photonics. *ACS Photonics* **11**, 1703–1714 (2024).
6. Chen, Z., Li, Z., Deng, Z., Liu, J. & Yu, S. An Optoelectronic Analog Ising Machine Enabling 2048-Spin and Low-Latency Calculations. in *Optical Fiber Communications Conference and Exhibition (OFC)* 1–3 (2023).

7. Chen, G. *et al.* Large scale optoelectronic Ising machine based on distributed parallel architecture. in *Conferenceon Lasers & Electro-Optics (CLEO*) JSS129_5 (2025).
8. S. Williams *et al.* Optimization of sparse matrix-vector multiplication on emerging multicore platforms. in *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing* 1–12 (2007).
9. Forney, G. Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Trans. Inf. Theory* **18**, 363–378 (1972).
10. Damodaran, R. *et al.* A 1.25GHz 0.8W C66x DSP Core in 40nm CMOS. in *International Conference on VLSI Design* 286–291 (2012).