DS-OPT: OPTIMIZATION-NATIVE REASONING WITH DYNAMICAL SYSTEM MACHINES

Anonymous authors
Paper under double-blind review

ABSTRACT

Many structured reasoning tasks, such as logic puzzles and constraint satisfaction, require solving combinatorial optimization problems over many interacting discrete variables. Neural networks are built from primitives such as matrix multiplication, which excel at representation learning but are not native mechanisms for combinatorial search, so neural models must grow in size or computation to emulate reasoning over global constraints. We propose DS-OPT, a neural reasoning framework that adds a Dynamical System Machine (DSM) as a layer of the network. The neural layers learn to construct, for each input, a multi-state energy whose low-energy states answer the task, and the DSM minimizes this energy through its native discrete dynamics, using noisy winner-take-all units that reduce to an Ising machine in the two-state case. A differentiable relaxation makes the layer trainable end to end, while inference always runs the discrete DSM. On Zebra puzzles, visual Sudoku and binary image completion, DS-OPT is more accurate than neural baselines of comparable size. On Zebra it solves up to 98.4% of the puzzles, against 82.6% for a Transformer with five times more parameters, at half its inference time and energy, with DSM costs estimated by circuit-level simulation. These results suggest that DSM hardware can complement neural networks as an efficient substrate for combinatorial reasoning.

1 INTRODUCTION

Modern neural networks, especially Transformer-based architectures, have become the dominant computational substrate for perception, language understanding, and decision making. However, structured reasoning remains difficult when a task requires satisfying global constraints over many interacting variables. In benchmarks such as Sudoku, logic puzzles, and planning, the model must resolve discrete constraints or search over valid configurations rather than simply produce locally plausible predictions (Sun et al., 2026). Scaling depth, width, recurrence, or test-time computation can improve performance (Wang et al., 2025; Jolicoeur-Martineau, 2025), but this asks the same neural primitive to emulate an optimization process through expensive neural transformations.

This difficulty reflects a mismatch between computation and problem structure. Transformers are highly effective at sequence modeling, representation learning, and information mixing, but many reasoning-intensive problems are more naturally viewed as constrained optimization over discrete state spaces, such as SAT (Massacci & Marraro, 2000; Mironov & Zhang, 2006; Lafitte et al., 2014), QUBO (Lu et al., 2024; Glover et al., 2022), Ising (Lucas, 2014; Takabatake et al., 2022), and other optimization formulations. Rather than forcing a single homogeneous neural architecture to handle perception, representation, and combinatorial optimization simultaneously, a more natural design is to introduce computational specialization. Similar to modular computation in biological intelligence, where different functional regions support different kinds of processing (Fodor, 1983; Kanwisher, 2010), AI systems may benefit from assigning optimization-heavy reasoning to specialized computational primitives (Jacobs et al., 1991; Hopfield & Tank, 1985; Kirkpatrick et al., 1983).

Dynamical System Machines (DSMs) provide a promising substrate for this role. A DSM implements a configurable energy landscape and performs optimization through autonomous physical relaxation, making it a native optimization primitive rather than a sequence of digital arithmetic operations. This sets it apart from a neural network at the root: a neural network approximates a mapping by fitting stacked matrix multiplications to data, whereas a discrete DSM is an energy

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

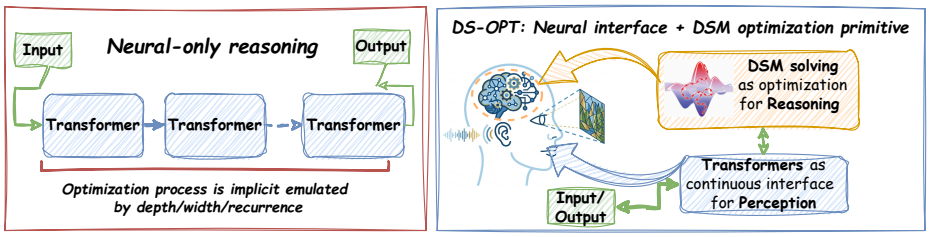


Figure 1: From neural-only reasoning to DS-OPT: neural networks for perception and interfacing, DSMs for reasoning and optimization.

model in which a compact set of couplings can encode a hard combinatorial problem and the physical dynamics search for its low-energy states. Solving NP-hard problems in this way is the classic use of Ising machines (Lucas, 2014; Hamerly et al., 2019), and prior DSM works have also shown promise for AI workloads (Wu et al., 2024; Liu et al., 2025b; Song et al., 2025). However, many AI-oriented DSM approaches design continuous quadratic or near-quadratic energy functions to better process real-valued features and simplify training. While useful for continuous transformations, such formulations can reduce the distinctive combinatorial structure of discrete DSMs; in stable quadratic regimes, the dynamics are closely related to solving continuous linear systems. Increasing expressivity by adding higher-order interactions or complex nonlinear circuits may recover some capability, but also increases hardware cost and implementation complexity. This gap motivates our central design idea: let neural networks handle perception and interfacing, and let a discrete DSM handle reasoning as native optimization.

We propose **DS-OPT**, a neural reasoning framework built around this division of labor. As Figure 1 illustrates, neural-only models must absorb reasoning into stacked Transformer computations. In DS-OPT, the neural network perceives the input and translates it into a problem for the DSM, and the DSM reasons by optimizing, reaching the answer as its low-energy state. Each part does what it is best at: neural networks perceive and interface, DSMs reason and optimize.

We compare DS-OPT with neural-only baselines and published methods on Zebra logic puzzles, visual Sudoku and image completion. DS-OPT is more accurate than neural baselines of comparable size, and on Zebra it outperforms a Transformer with five times more parameters at half its inference time and energy. Our main contributions are:

- We propose DS-OPT, a neural reasoning framework built on a division of labor: neural layers perceive and interface, a DSM layer reasons and optimizes, and together they reason better than the network alone.
- We develop a trainable multi-state DSM layer with differentiable relaxation and input-dependent energy construction, which runs on winner-take-all hardware and reduces to an Ising DSM in the two-state case.
- We demonstrate improved reasoning accuracy on Zebra puzzles, visual Sudoku, and image completion, with fewer parameters, less inference time and less energy than neural-only baselines of similar accuracy.

2 Background and Motivation

2.1 Continuous and Discrete DSMs

Dynamical System Machines (DSMs) are energy-based computing substrates that solve optimization problems through autonomous physical dynamics. Instead of executing a sequence of digital arithmetic instructions, a DSM is configured with an energy function whose low-energy states encode desirable solutions; once initialized, its physical state evolves toward energetically favorable configurations, a process known as natural annealing. Figure 2a shows the hardware abstraction we use: state variables are physical nodes, such as capacitor voltages or oscillator phases, and tunable coupling units realize the interactions between them.

Early DSMs, in particular Ising machines (Afoakwa et al., 2021; Inagaki et al., 2016; Lo et al., 2023), operate in a discrete regime. Each node settles into one of two states, and the machine minimizes the Ising energy of Eq. (3) in Section 3.1, an NP-hard objective whose rugged landscape suits reasoning

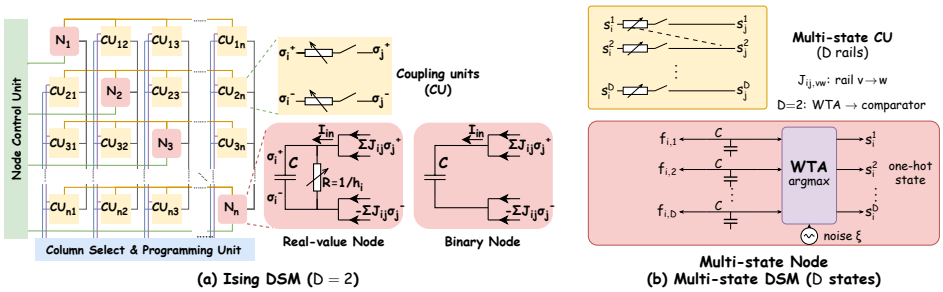


Figure 2: DSM hardware. (a) Ising DSM: nodes (N_i) and coupling units (CU) on a crossbar; each CU connects two nodes through tunable resistors, and a node integrates the coupling current on a capacitor (with a leak resistor for real-valued nodes). (b) Multi-state DSM: D rails per node; a CU couples rail v of node i to rail w of node j with weight $J_{ij,vw}$ (dashed: a cross-state term), and each node selects its state with a noisy winner-take-all (WTA) circuit, which for $D = 2$ reduces to a comparator.

over many interacting constraints. Many AI workloads, however, operate on continuous features, and AI-oriented DSMs such as NPGL (Wu et al., 2024) and its follow-ups (Song et al., 2025; Liu et al., 2025a) relax the binary state into a continuous one governed by a quadratic energy. In the stable regime this energy has a single minimum, and the dynamics effectively solve a linear system (Appendix A.2). Continuous DSMs are therefore efficient for real-valued transformations, but they give up the combinatorial search that makes a discrete DSM useful for reasoning, and recovering expressivity through higher-order interactions or nonlinear node circuits adds circuit complexity, noise sensitivity and scaling difficulty. DS-OPT instead keeps the DSM in its native discrete regime and leaves continuous processing to the neural layers.

2.2 Multi-State DSMs and Potts Machines

Many reasoning problems are categorical: a Sudoku cell holds one of nine digits, and an attribute value in a logic puzzle belongs to one of several houses. On an Ising machine such a variable needs a one-hot encoding with D spins and a penalty that keeps exactly one of them on (Lucas, 2014). The encoding multiplies the number of spins and couplings, and it makes the landscape harder to search, because changing a value has to pass through states that violate the encoding. The Potts model, the D-state generalization of the Ising model, represents such variables directly. Potts machines have been built with CMOS-compatible stochastic annealers based on single-photon avalanche diodes, running both Ising and Potts annealing on one platform (Whitehead et al., 2023), and designed with coupled CMOS ring oscillators whose phase, locked by sub-harmonic injection, takes one of several values (Gonul & Taskin, 2024; 2025; Cheng & Lin, 2025), as well as with optical parametric oscillators that have three stable phases (Honari-Latifpour & Miri, 2020). Accelerator frameworks for multi-state spin problems have also been proposed (Garg & Salahuddin, 2025).

The key hardware difference from an Ising machine is how a node chooses its state. An Ising node compares its local field with zero, whereas a multi-state node must select the largest of D competing inputs, a winner-take-all (WTA) operation (Figure 2b). WTA is a standard analog building block that can be built in CMOS with interconnect growing only linearly in the number of inputs (Lazzaro et al., 1988), and adding noise before the comparison turns the selection into sampling. With $D = 2$ the WTA reduces to the comparator of an Ising machine, so an Ising machine is the 2-state special case. We design the DSM of DS-OPT after this class of machines (Section 3.1): its nodes are noisy WTA units, it keeps the discrete search of Ising machines, and it represents categorical variables without an encoding penalty. When only binary hardware is available, a multi-state energy can still be mapped exactly to a one-hot Ising machine, at the cost of D times more spins (Appendix D).

3 Methodology

We now introduce the core of DS-OPT: a DSM-compatible optimization layer that turns the energy minimization performed natively by DSM hardware into a trainable reasoning primitive. The layer builds a multi-state energy from neural features and returns the low-energy state reached by the

DSM dynamics. It can therefore sit on top of Transformer layers, or other neural front ends, and be trained with them as a single hybrid model, in which the network handles continuous representations and the DSM carries out the combinatorial search in its native discrete regime. Section 3.1 defines the multi-state DSM, its hardware dynamics and the relaxation that makes it trainable; Section 3.2 describes how the layer constructs its energy from neural features; and Section 3.3 explains how the resulting model is trained and used for inference.

3.1 Multi-State DSM Dynamics and Their Differentiable Relaxation

The DSM holds N discrete variables, each taking one of D states, $x_i \in \{1, \dots, D\}$, written as a one-hot vector $\mathbf{s}_i \in \{0,1\}^D$. For each input, the network constructs the energy

$$\mathcal{H}(\mathbf{s}; \mathbf{J}, \mathbf{h}) = -\sum_{i=1}^{N} \mathbf{h}_i^\top \mathbf{s}_i - \frac{1}{2} \sum_{i \neq j} \mathbf{s}_i^\top \mathbf{J}_{ij}\, \mathbf{s}_j, \tag{1}$$

where $\mathbf{h}_i \in \mathbb{R}^D$ is the field on variable i and $\mathbf{J}_{ij} \in \mathbb{R}^{D \times D}$ couples the states of variables i and j, with $\mathbf{J}_{ji} = \mathbf{J}_{ij}^\top$ and no self-coupling. For simplicity we use a common D; a variable with fewer states is padded to D with the extra states masked out. The lowest-energy assignment is the answer. With the other variables fixed, the energy depends on the state of variable i only through its local field

$$\mathbf{f}_i = \mathbf{h}_i + \sum_{j \neq i} \mathbf{J}_{ij}\, \mathbf{s}_j \ \in \mathbb{R}^D, \tag{2}$$

and the state with the largest component of $\mathbf{f}_i$ has the lowest energy.

The Ising model is the 2-state case. With $D = 2$, a spin $\sigma_i \in \{-1, +1\}$ encoded as $\mathbf{s}_i = \left(\frac{1+\sigma_i}{2}, \frac{1-\sigma_i}{2}\right)^\top$, fields $\mathbf{h}_i = b_i\,(1, -1)^\top$ and couplings $\mathbf{J}_{ij} = J_{ij} \left(\begin{smallmatrix} 1 & -1 \\ -1 & 1 \end{smallmatrix}\right)$, Eq. (1) reduces to

$$\mathcal{H}(\boldsymbol{\sigma}; J, \mathbf{b}) = -\frac{1}{2} \boldsymbol{\sigma}^\top J \boldsymbol{\sigma} - \mathbf{b}^\top \boldsymbol{\sigma}, \qquad \boldsymbol{\sigma} \in \{-1, +1\}^N, \tag{3}$$

with $J \in \mathbb{R}^{N \times N}$ symmetric and zero on the diagonal. Same-state couplings $\mathbf{J}_{ij} = J_{ij}\, I_D$, which reward or penalize two variables for taking the same state, give the Potts model, whose hardware needs a single $N \times N$ coupling matrix shared by all D states. Minimizing Eq. (1) is NP-hard, as it includes Ising optimization and graph coloring.

A multi-state DSM implements this energy with one winner-take-all (WTA) unit per variable. The unit of variable i sums the coupling contributions into the local field and selects the state whose field, perturbed by analog noise, is largest:

$$x_i \leftarrow \underset{v \in \{1, \dots, D\}}{\arg\max} \left(f_{i,v} + T_t\, \xi_{i,v}\right), \qquad \xi_{i,v} \sim \mathcal{N}\left(0, \tfrac{\pi^2}{6}\right), \tag{4}$$

where the noise amplitude T_t is lowered over the anneal. Variables are updated asynchronously, one at a time or in random blocks, and a sweep updates every variable once. For $D = 2$ the WTA becomes a noisy comparator, $\sigma_i \leftarrow \operatorname{sign}\left(\sum_j J_{ij}\sigma_j + b_i + T_t\, \zeta_i\right)$ with $\zeta_i = (\xi_{i,1} - \xi_{i,2})/2$. The update samples rather than descends: with Gumbel noise in place of the Gaussian, the selected state would follow $\operatorname{softmax}(\mathbf{f}_i / T_t)$ exactly, the Boltzmann distribution of variable i given the others (the Gumbel-max identity (Maddison et al., 2014)), and Gaussian noise of the same variance is what analog circuits naturally provide; with Gaussian noise, or when a block of coupled variables is updated at once, the update approximates rather than equals Gibbs sampling. An update thus needs only summed coupling currents, noise and a comparison, with no softmax; the only scaling is a one-time normalization of the energy when it is loaded (Section 3.3).

The WTA has no useful gradient, so the DSM cannot be trained through it directly, yet the training surrogate must stay faithful to the discrete dynamics, so that a learned energy gives the same answers under discrete annealing. During training only, we replace each one-hot state with a point on the probability simplex and simulate a finite number of leaky-integrator steps:

$$\mathbf{m}_i^{(k)} = \operatorname{softmax}\left(\mathbf{u}_i^{(k)} / T_k\right), \qquad \widetilde{\mathbf{f}}_i^{(k)} = \mathbf{h}_i + \sum_{j \neq i} \mathbf{J}_{ij}\, \mathbf{m}_j^{(k)},$$

$$\mathbf{u}_i^{(k+1)} = (1 - \lambda)\, \mathbf{u}_i^{(k)} + \eta\, \widetilde{\mathbf{f}}_i^{(k)} / r^{(k)}, \tag{5}$$

for $k = 0, \ldots, K_{\mathrm{r}} - 1$, starting from $\mathbf{u}_i^{(0)} = \mathbf{0}$, with leak λ, step size η and a temperature that decreases geometrically from $T_0 = T_{\mathrm{init}}$ to $T_{K_{\mathrm{r}}-1} = T_{\mathrm{final}}$. The output of the relaxation is $\mathbf{m}_i^{(K_{\mathrm{r}})} = \mathrm{softmax}\left(\mathbf{u}_i^{(K_{\mathrm{r}})}/T_{\mathrm{final}}\right)$. Here $\mathbf{u}_i$ plays the role of the integrator voltages and the softmax is a smoothed WTA. The factor $r^{(k)}$ fixes the scale of the energy: it is the root mean square of $\widetilde{\mathbf{f}}^{(k)}$ over all variables and states of an instance, or $r^{(k)} = 1$ when the couplings are normalized directly, as in our Sudoku models; without it the energy grows during training until the softmax saturates and the gradient vanishes. For $D = 2$, $\widetilde{\sigma}_i = m_{i,1} - m_{i,2} = \tanh\big((u_{i,1} - u_{i,2})/(2T_k)\big)$ recovers the $\tanh$ relaxation of Ising dynamics.

The relaxation is consistent with the discrete model because Eq. (1), extended to simplex-valued states, is linear in each $\mathbf{m}_i$ when the others are fixed, so its minimum over the product of simplices is attained at a one-hot vertex, and the softmax approaches the WTA as $T_k \to 0$. The relaxed trajectory need not match the physical one; DSM solving is stochastic, and what matters is that the two settle on the same low-energy states. Section 3.3 describes how training ensures this, and Appendix B verifies it.

3.2 The DSM Layer

DS-OPT treats the DSM as a layer of the network rather than as an external solver. Just as an attention layer computes its mixing weights from its input, the DSM layer computes the fields and couplings of Eq. (1) from the features of the preceding layers and lets its dynamics settle this energy. Each DSM variable stands for a variable of the task, such as a Sudoku cell, an attribute value of a logic puzzle or a pixel, and its settled state is the prediction for it. The preceding layers are a Transformer encoder for Zebra puzzles and image completion and a LeNet for visual Sudoku. The neural layers and energy projections are trained jointly through the relaxed dynamics, so the loss on the DSM state shapes both the features and the energy, and no part of the energy is hand-written; the one exception, the shared Sudoku couplings, is learned beforehand from symbolic puzzles (Section 3.3).

Keeping the DSM on the task variables also keeps it compact, which matters because scalability is a fundamental concern of DSM hardware. A dense energy over N variables with D states has up to $O(N^2D^2)$ coupling values, and giving every neural feature channel its own spin would multiply this further, while partitioning a dense energy across smaller devices (Song et al., 2024; Sharma et al., 2022) weakens the interactions between partitions. In DS-OPT, feature processing stays in the neural layers; our largest instance has 2,304 binary variables, of which at most 400 are free, and fits in one DSM core.

Let $\mathbf{z}_i \in \mathbb{R}^d$ denote the features of variable i. In general the field is a linear projection, $\mathbf{h}_i = W_h\mathbf{z}_i$, with states that a variable cannot take masked out. The couplings depend on how the rules of the task vary, and we use three forms.

Shared couplings for fixed rules. When the rules are the same for every instance, as in Sudoku, the couplings should not depend on the input. We use same-state couplings $\mathbf{J}_{ij} = J_{ij}I_D$ computed from learned position embeddings $\mathbf{e}_i$:

$$A = \frac{QK^\top}{\sqrt{d_q}}, \qquad \overline{A} = \frac{1}{2}\big(A + A^\top\big) \odot \big(\mathbf{1}\mathbf{1}^\top - I\big), \qquad J = \gamma\,\frac{\overline{A}}{\|\overline{A}\|_{\mathrm{rms}}}, \tag{6}$$

where the rows of Q and K are linear projections of the $\mathbf{e}_i$, $\|\cdot\|_{\mathrm{rms}}$ is taken over the off-diagonal entries and γ is a fixed scale. As in attention, two positions are strongly coupled when their projected embeddings are compatible; symmetrization makes the interaction reciprocal, the mask removes self-couplings, and fixing the scale removes the freedom to trade the magnitude of the couplings against that of the fields. Only the fields vary across instances. In visual Sudoku the field of a given cell is $\mathbf{h}_i = \tau \log \mathbf{p}_i$, where $\mathbf{p}_i$ are the LeNet's digit probabilities for the cell image and τ is learned; blank cells receive no field.

Instance-specific couplings for varying rules. When every instance has its own rules, as in Zebra puzzles, the couplings must come from the input. The Transformer reads the clue tokens together with one token per variable, and every pair of variable features is projected to a full coupling table:

$$P_{ij} = \mathrm{MLP}\left(W_q\mathbf{z}_i + W_k\mathbf{z}_j\right) \in \mathbb{R}^{D\times D}, \qquad \mathbf{J}_{ij} = \frac{1}{2}\big(P_{ij} + P_{ji}^\top\big), \tag{7}$$

Algorithm 1 Training and inference with a DS-OPT model (sg: stop-gradient; fixed Sudoku couplings are not updated)

procedure TRAINSTEP(input $\mathcal{X}$, targets $\mathbf{y}$ on supervised variables $\mathcal{M}$)
  $(\mathbf{h}, \mathbf{J}) \leftarrow E_\phi(F_\theta(\mathcal{X}))$; $\mathbf{u}^{(0)} \leftarrow \mathbf{0}$ ▷ neural layers, energy projections
  for $k = 0$ **to** $K_\mathrm{r} - 1$ **do** ▷ relaxed dynamics, Eq. (5)
    $\mathbf{m}^{(k)} \leftarrow \mathrm{softmax}(\mathbf{u}^{(k)}/T_k)$
    completion only: $\mathbf{m}^{(k)} \leftarrow \mathbf{m}^{(k)} + \mathrm{sg}(\widehat{\mathbf{s}}^{(k)} - \mathbf{m}^{(k)})$, $\widehat{\mathbf{s}}^{(k)} \sim \mathbf{m}^{(k)}$ ▷ discrete forward
    $\mathbf{u}^{(k+1)} \leftarrow (1-\lambda)\,\mathbf{u}^{(k)} + \eta\,\widetilde{\mathbf{f}}^{(k)}/r^{(k)}$
  end for
  $\mathcal{L} \leftarrow -\frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \log m_{i,y_i}^{(K_\mathrm{r})}$ with $\mathbf{m}^{(K_\mathrm{r})} = \mathrm{softmax}(\mathbf{u}^{(K_\mathrm{r})}/T_\mathrm{final})$; update θ, ϕ
end procedure
procedure INFER(input $\mathcal{X}$)
  $(\mathbf{h}, \mathbf{J}) \leftarrow E_\phi(F_\theta(\mathcal{X}))$, normalized by $1/r$ on the DSM
  run K parallel anneals from random states, each S sweeps of Eq. (4) with falling T_t
  return the lowest-energy state found
end procedure

which satisfies $\mathbf{J}_{ji} = \mathbf{J}_{ij}^\top$ by construction. A full table can express relations such as "in the same house", "immediately left of" or "next to". A relation among three variables, such as "in between", cannot be written with pairwise terms, so we add one binary auxiliary variable per clue; its features come from the clue token, it is coupled to the task variables through the same projection, and auxiliary variables are not coupled to one another.

Local couplings for implicit rules. In image completion no rule can be written down, and neighbouring pixels interact most strongly. Each pixel is a binary variable (Eq. (3)) with field $b_i = \mathbf{w}_b^\top \mathbf{z}_i$. Couplings are defined for a fixed set $\mathcal{O}$ of 27 displacements within a 17×17 window, so each pixel interacts with up to 54 others, and a pixel pair at displacement $o \in \mathcal{O}$ has coupling $J_{ij} = \mathbf{w}_o^\top [\mathbf{z}_i; \mathbf{z}_j] + c_o$. Observed pixels are clamped to their values.

In all three forms the energy is an internal quantity of the network, recomputed for every input like attention weights: the neural layers learn to express the structure of each instance in it, and the DSM dynamics carry out the combinatorial reasoning it requires.

3.3 TRAINING AND INFERENCE

Training and inference run the same network and construct the same energy; they differ only in how the DSM layer settles it (Algorithm 1).

Training. The DSM layer runs the relaxed dynamics of Eq. (5) for K_r steps, and the loss is the cross-entropy between the targets and the relaxed output $\mathbf{m}^{(K_\mathrm{r})}$ on the supervised variables $\mathcal{M}$: all task variables in Zebra, the blank cells in visual Sudoku and the masked pixels in image completion. For binary variables we use the equivalent logistic loss on a learned scale of the relaxed spin. The model is trained by backpropagation through the relaxation, so the gradient reaches the neural layers only through the DSM dynamics. The one exception is the shared Sudoku couplings, learned beforehand on symbolic Sudoku puzzles and kept fixed while the LeNet and the field scale are trained.

Two measures keep the relaxation faithful to the discrete dynamics. First, T_final is held at a moderate value early in training, where a hot relaxation keeps the gradient alive, and lowered geometrically over the second half, so that the relaxed output settles on the states the discrete anneal finds. We use this schedule for Zebra (T_final from 1.0 to 0.1) and image completion (from 0.5 to 0.05); for visual Sudoku a fixed $T_\mathrm{final} = 0.7$ already gives matching results. Second, for image completion the relaxation uses discrete states in the forward pass: at every step the state is sampled from $\mathbf{m}^{(k)}$, and the gradient is passed through as if $\mathbf{m}^{(k)}$ had been used (straight-through), so the network trains on the states it meets at inference.

Inference. The relaxation is not used for prediction; all reported results come from discrete annealing with the WTA rule of Eq. (4). K anneals run in parallel from random states, each for S sweeps with T_t lowered geometrically, and the lowest-energy state is returned. When the drive is normalized during training, the DSM applies the same normalization $1/r$ to the energy of each instance, so that a given noise amplitude has the same meaning in training and inference. The annealing process is validated by circuit-level mixed-signal simulation.

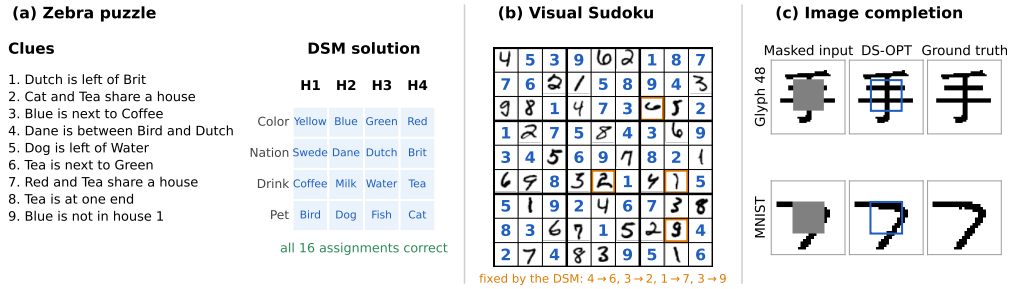


Figure 3: DS-OPT on test instances. (a) A Zebra puzzle solved by the DSM (attribute names for display only). (b) Visual Sudoku: blue digits are the DSM's answers; the four digits misread by the LeNet (orange) are corrected. (c) Image completion: the DSM fills in the masked region (blue box).

Table 1: Zebra puzzles, 100k test puzzles. Inference time and energy are for the whole test set (GPU: time $\times$ 250 W; DS-OPT includes the DSM). DS-OPT and the 30-epoch Transformer: best of 3 and 2 seeds. *1.5M training puzzles (5$\times$ ours). †Equivalent re-implementation.

Model	Params	Acc. (%)	Training	Inference	Energy
DS-OPT (ours)	1.33M	**98.4**	1.1 h	5.1 s	1.3 kJ
without couplings (J=0)	1.33M	2.6	—	5.1 s	1.3 kJ
Transformer, same interface (30 ep)	1.28M	76.0	0.34 h	3.7 s	0.91 kJ
Transformer, same interface (100 ep)	1.28M	79.9	1.1 h	3.7 s	0.91 kJ
Transformer, 5$\times$ larger (30 ep)	6.66M	82.6	0.86 h	10 s	2.5 kJ
AR Transformer (Shah et al., 2024)	42M	79.4	51 h†	5.1 min†	77 kJ†
AR + solver traces (Shah et al., 2024)	42M	92.0	51 h†	—	—
Masked diffusion* (Kim et al., 2025)	19M	98.5	—	—	—
Glauber-UL2* (Kathuria & Kumar, 2026)	21M	98.7	—	—	—
Flow Reasoning Model* (Helbling et al., 2026)	25M	100.0	—	—	—

4 Evaluation

We evaluate DS-OPT on three tasks with different kinds of rules: rules that change with every instance (Zebra puzzles), fixed rules over perceptual input (visual Sudoku), and implicit rules that cannot be written down (binary image completion). All results come from discrete WTA annealing with K parallel restarts (budgets in Appendix C.3). Our main baseline is a Transformer with the same interface, encoder, data and number of epochs, whose head predicts every variable in one forward pass; we also report larger and longer-trained versions of it and published results on the same benchmarks. DS-OPT with all couplings removed (J=0) tests whether the reasoning happens in the DSM. For every model we report the number of parameters, the task metric, the training time, and the inference time and energy on the full test set. Models run on NVIDIA A100 GPUs, whose energy is inference time multiplied by 250 W; the time and energy of the DSM come from circuit-level simulation in Cadence Virtuoso with a 45 nm process (Appendix C.4).

4.1 Zebra Puzzles: Rules That Change with Every Instance

Zebra puzzles ask for an assignment of attribute values to houses that satisfies a set of clues. We use the benchmark of Shah et al. (2024): 3–6 houses, 3–6 attributes, seven clue types and a unique solution; the metric is the fraction of puzzles solved completely. We train on 300k of the released puzzles, using only (clues, solution) pairs without the solver traces used by some baselines, and test on all 100k test puzzles. Unlike Sudoku, every Zebra puzzle has its own rules, so the model must turn the clues of each puzzle into its constraints: the Transformer constructs an energy over the house of each attribute value ($D \leq 6$), with one binary auxiliary variable per clue so that three-way clues can be expressed (Section 3.2).

DS-OPT solves up to 98.4% of the puzzles (96.5% on average over three seeds), against 76.0% for the Transformer with the same interface (Table 1). Training the Transformer more than three times longer (79.9%) or making it five times larger (82.6%) still leaves it about 14 points behind the DS-OPT average, and the larger Transformer needs twice the inference time and energy of DS-OPT. Without couplings DS-OPT collapses to 2.6%, so the answers come from minimizing the

Table 2: Visual Sudoku, 1k test boards (all 81 cells correct); time and energy as in Table 1. DS-OPT: mean of 4 seeds. *One trained model, which solves 97.2% when reading and reasoning run jointly. °6.85M unrolled. +Includes 13M pre-trained perception.

Model	Params	Acc. (%)	Training	Inference	Energy
DS-OPT (ours, LeNet + DSM)	1.23M	**97.6**	3.6 min	0.13 s	33 J
LeNet readings fixed, then DSM*	1.23M	61.1	—	—	—
Neural models					
Recurrent Transformer (Yang et al., 2023)	0.70M°	93.5	2.1 h	0.27 s	68 J
IRED (Du et al., 2024)	≥44.9M	98.3	—	—	—
Neuro-symbolic models					
NeSy Programming (Li et al., 2023)	—	95.5	—	—	—
SATNet (Topan et al., 2021)	14M+	64.8	—	—	—
SATNet (Chang et al., 2020)	1.05M	0.0	—	—	—

Table 3: Binary image completion: pixel accuracy on the masked region; time and energy as in Table 1 (200 / 200 / 1k test images). All-background predictor: 64.4 / 62.1 / 66.1.

Dataset	Model	Params	Acc. (%)	Training	Inference (s)	Energy (J)
Glyph 24	DS-OPT (ours)	0.87M	**71.1**	1.4 h	0.0107	2.68
	$J{=}0$	0.87M	64.4	—	0.0107	2.68
	Transformer	0.93M	68.4	0.09 h	0.0110	2.75
Glyph 48	DS-OPT (ours)	0.91M	**87.2**	1.3 h	0.0354	8.85
	$J{=}0$	0.91M	64.8	—	0.0354	8.85
	Transformer	0.92M	79.8	0.09 h	0.0376	9.40
MNIST	DS-OPT (ours)	0.90M	**79.8**	1.3 h	0.260	64.9
	$J{=}0$	0.90M	66.1	—	0.260	64.9
	Transformer	0.96M	77.9	0.13 h	0.269	67.1

constructed energy, not from the per-variable fields. DS-OPT also outperforms the autoregressive Transformers of Shah et al. (2024), including the variant trained on solver traces. Masked-diffusion and flow models reach 98.5–100% with 14–19× more parameters and five times more training puzzles. Figure 3a shows a solved puzzle.

4.2 Visual Sudoku: Perception and Reasoning in One Model

In visual Sudoku (Wang et al., 2019) every given digit is an MNIST image (9k training and 1k test boards). We follow the *ungrounded* protocol (Chang et al., 2020; Yang et al., 2023): the digit images carry no labels and the loss covers only the blank cells, so the digit reader is trained only through reasoning. As in prior work the reader is a LeNet, trained from scratch through the relaxation; its digit probabilities set the fields of a 9-state DSM over the 81 cells, whose couplings encode the Sudoku rules and are learned beforehand on symbolic puzzles (Appendix C.2). A board counts as solved only if all 81 cells are correct.

DS-OPT solves 97.6% of the boards (Table 2), above the Recurrent Transformer (93.5%), a model of comparable size, and the neuro-symbolic method of Li et al. (2023) (95.5%), which uses an exact MaxSAT solver, although our criterion is stricter than theirs, with little variation across the four seeds. IRED reports 98.3% with at least 36× more parameters, without a description of its evaluation protocol. Reasoning also repairs perception. On one trained model, the LeNet reads 98.6% of the given digits correctly, but one misread digit is enough to make a board wrong: taking its readings as fixed and solving the resulting puzzles with the same DSM solves only 61.1% of the boards. When perception and reasoning run together, the DSM overrides 94.7% of the misreadings, which conflict with the rest of the board, and the same model solves 97.2% (Figure 3b).

4.3 Binary Image Completion: Rules That Cannot Be Written Down

In the previous tasks the rules could, in principle, be written by hand; here they cannot. A square covering about 40% of the side of a binary image is masked and must be filled in, on two synthetic glyph datasets (24×24 and 48×48 pixels) and binarized MNIST, scored by pixel accuracy on

the masked region. Each pixel is a binary DSM variable with couplings over 27 displacements, constructed per image by the Transformer. DS-OPT improves over the Transformer of the same size on all three datasets (Table 3). Without couplings it falls close to the all-background baseline, so the gain comes from learned pixel interactions: the network learns an energy whose low-energy states are plausible shapes, although no rule for them is available (Figure 3c).

5 Related Work

Dynamical-system, Ising and Potts machines. Ising machines relax pairwise-coupled binary spins toward low-energy states and have been built with resistively coupled capacitors (Afoakwa et al., 2021), optical parametric oscillators (Inagaki et al., 2016; Hamerly et al., 2019) and coupled CMOS oscillators (Moy et al., 2022; Lo et al., 2023; Cılasun et al., 2025); Potts machines extend them to multi-state spins with single-photon avalanche diodes (Whitehead et al., 2023), coupled ring oscillators (Gonul & Taskin, 2024; 2025), other coupled oscillators (Cheng & Lin, 2025) and optical oscillators (Honari-Latifpour & Miri, 2020). The use of such dynamics for optimization goes back to Hopfield–Tank (Hopfield & Tank, 1985) and Potts neural networks (Peterson & Söderberg, 1989), whose mean-field dynamics resemble our training relaxation. Real-valued spin dynamics have also served graph learning (Wu et al., 2024; Song et al., 2024), function learning (Liu et al., 2025b), training (Liu et al., 2025a) and language processing (Song et al., 2025), with continuous states and fixed encodings; DS-OPT instead makes a discrete DSM a trainable layer whose energy is constructed from neural features for every input.

Differentiable optimization and energy-based learning. OptNet (Amos & Kolter, 2017) differentiates through a convex program, and combinatorial solvers can be differentiated by interpolating (Vlastelica et al., 2020) or perturbing (Berthet et al., 2020) their inputs, but the solver remains a digital routine on a fixed problem form. Structured prediction energy networks (Belanger & McCallum, 2016) minimize a learned energy by gradient descent in a continuous relaxation; our training relaxation is closest to them, combined with sampled discrete states and straight-through gradients (Bengio et al., 2013; Jang et al., 2017), but the energy it trains is the one a DSM minimizes natively. Equilibrium propagation (Scellier & Bengio, 2017) and physical neural networks (Wright et al., 2022) train physical systems directly, whereas DS-OPT trains on GPUs and leaves inference to the hardware.

Structured reasoning models. Neural models tackle Sudoku and logic puzzles with architectural bias and iterative computation: recurrent Transformers (Yang et al., 2023), recursive reasoning modules (Wang et al., 2025; Jolicoeur-Martineau, 2025), autoregressive models trained on solver traces (Shah et al., 2024), masked diffusion (Kim et al., 2025) and energy-based refinement (Du et al., 2024). Neuro-symbolic methods embed a solver: a differentiable MaxSAT layer (Wang et al., 2019; Topan et al., 2021), a learned constraint program solved exactly (Li et al., 2023), or learned pairwise costs solved by a weighted constraint solver (Defresne et al., 2023). DS-OPT instead learns an energy for each input and minimizes it with the native, parallel dynamics of a DSM.

6 Conclusion

This paper presents DS-OPT, a neural reasoning framework that adds a discrete, multi-state DSM layer to neural networks. The DSM stays in its native regime of discrete energy minimization, built from noisy winner-take-all units that reduce to an Ising machine for two states, while the neural layers learn to construct, for every input, the energy whose low-energy states answer the task; a differentiable relaxation used only in training makes the model trainable end to end, and inference runs discrete annealing. On Zebra puzzles, visual Sudoku and binary image completion, DS-OPT is more accurate than both same-interface Transformers and published models of comparable size, and removing the couplings erases the gain. Scaling the Transformer does not close the gap: on Zebra, DS-OPT with 1.33M parameters solves up to 98.4% of the puzzles against 82.6% for a Transformer five times larger, at half its inference time and energy, and the DSM accounts for less than 0.1% of that time. Our estimates rely on circuit-level simulation, and our largest instance (2,304 binary variables, at most 400 free) fits in a single core; larger energies across several cores and training on the hardware are natural next steps. These results suggest that specialized physical primitives can take over the combinatorial part of reasoning from ever-larger neural networks.

Reproducibility Statement

The DS-OPT model, its relaxed training dynamics and its discrete inference procedure are specified in Section 3 and Algorithm 1, and the three forms of the energy in Section 3.2. Appendix B gives complete proofs, with their assumptions, of the consistency between the relaxed and discrete models and of the relationship between noisy WTA updates and Gibbs sampling, and Appendix D proves the exactness of the mapping to binary Ising hardware. The data processing and the masks are described in Appendix C.1. Architectures, hyperparameters, annealing budgets and DSM sizes are listed in Appendix C.2 and Tables 5 and 6, and the protocol for measuring time, energy and parameters, including the re-implementation used to time the JAX-based baseline of Shah et al. (2024), in Appendix C.4. Results are reported on the full test sets; where several seeds were run, the table captions give their number.

AI Use Statement

In this work, we used generative AI tools to aid and polish writing and to assist with retrieval and discovery of relevant literature. We did not use generative AI tools for other tasks requiring disclosure. All AI-assisted content was reviewed by the authors, and identified literature was verified against the original sources. We take responsibility for the final content of this work.

References

Richard Afoakwa, Yiqiao Zhang, Uday Kumar Reddy Vengalam, Zeljko Ignjatovic, and Michael Huang. BRIM: Bistable resistively-coupled Ising machine. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 749–760. IEEE, 2021.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, 2017.

David Belanger and Andrew McCallum. Structured prediction energy networks. In *International Conference on Machine Learning*, 2016.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems*, 2020.

Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3):179–195, 1975.

Oscar Chang, Lampros Flokas, Hod Lipson, and Michael Spranger. Assessing SATNet's ability to solve the symbol grounding problem. In *Advances in Neural Information Processing Systems*, 2020.

Yi Cheng and Zongli Lin. From Ising to Potts: Physics-inspired Potts machines of coupled oscillators for low-energy sampling and combinatorial optimization. *arXiv preprint arXiv:2507.18379*, 2025.

Hüsrev Cılasun, William Moy, Ziqing Zeng, Tahmida Islam, Hao Lo, Alex Vanasse, Megan Tan, Mohammad Anees, Ramprasath S, Abhimanyu Kumar, Sachin S. Sapatnekar, Chris H. Kim, and Ulya R. Karpuzcu. A coupled-oscillator-based Ising chip for combinatorial optimization. *Nature Electronics*, 8(6):537–546, 2025. doi: 10.1038/s41928-025-01393-3.

Marianne Defresne, Sophie Barbe, and Thomas Schiex. Scalable coupling of deep learning with logical reasoning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, 2023. arXiv:2305.07617.

Yilun Du, Jiayuan Mao, and Joshua B. Tenenbaum. Learning iterative reasoning through energy diffusion. In *International Conference on Machine Learning*, 2024.

Jerry A. Fodor. *The Modularity of Mind*. MIT Press, Cambridge, MA, 1983.

Chirag Garg and Sayeef Salahuddin. Efficient optimization accelerator framework for multi-state spin Ising problems. *Nature Communications*, 16:9601, 2025. doi: 10.1038/s41467-025-64625-2.

Fred Glover, Gary Kochenberger, Rick Hennig, and Yu Du. Quantum bridge analytics I: a tutorial on formulating and using QUBO models. *Annals of Operations Research*, 314(1):141–183, 2022.

Yilmaz Ege Gonul and Baris Taskin. Multi-phase coupled CMOS ring oscillator based Potts machine. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–9, 2024. doi: 10.1145/3676536.3676720.

Yilmaz Ege Gonul and Baris Taskin. A multi-stage Potts machine based on coupled CMOS ring oscillators. In *Design, Automation and Test in Europe Conference (DATE)*, 2025. arXiv:2504.11376.

Ryan Hamerly, Takahiro Inagaki, Peter L. McMahon, Davide Venturelli, Alireza Marandi, Tatsuhiro Onodera, Edwin Ng, Carsten Langrock, Kensuke Inaba, Toshimori Honjo, Koji Enbutsu, Takeshi Umeki, Ryoichi Kasahara, Shoko Utsunomiya, Satoshi Kako, Ken-ichi Kawarabayashi, Robert L. Byer, Martin M. Fejer, Hideo Mabuchi, Dirk Englund, Eleanor Rieffel, Hiroki Takesue, and Yoshihisa Yamamoto. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Science Advances*, 5(5):eaau0823, 2019. doi: 10.1126/sciadv.aau0823.

Alec Helbling, Andrey Bryutkin, Mauro Martino, Duen Horng Chau, Nima Dehmamy, and Hendrik Strobelt. Flow reasoning models: Turning flows into efficient recurrent reasoners. *arXiv preprint arXiv:2606.29150*, 2026.

Mostafa Honari-Latifpour and Mohammad-Ali Miri. Optical Potts machine through networks of three-photon down-conversion oscillators. *Nanophotonics*, 9(13):4199–4205, 2020. doi: 10.1515/nanoph-2020-0256.

John J. Hopfield and David W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52(3):141–152, 1985. doi: 10.1007/BF00339943.

Takahiro Inagaki, Yoshitaka Haribara, Koji Igarashi, Tomohiro Sonobe, Shuhei Tamate, Toshimori Honjo, Alireza Marandi, Peter L. McMahon, Takeshi Umeki, Koji Enbutsu, Osamu Tadanaga, Hirokazu Takenouchi, Kazuyuki Aihara, Ken-ichi Kawarabayashi, Kyo Inoue, Shoko Utsunomiya, and Hiroki Takesue. A coherent Ising machine for 2000-node optimization problems. *Science*, 354(6312):603–606, 2016. doi: 10.1126/science.aah4243. URL `https://www.science.org/doi/abs/10.1126/science.aah4243`.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks, 2025. URL `https://arxiv.org/abs/2510.04871`.

Nancy Kanwisher. Functional specificity in the human brain: A window into the functional architecture of the mind. *Proceedings of the National Academy of Sciences*, 107(25):11163–11170, 2010. doi: 10.1073/pnas.1005062107.

Tarun Kathuria and Sachin Kumar. Leveraging pretrained language models as energy functions for Glauber dynamics text diffusion. *arXiv preprint arXiv:2605.04291*, 2026.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *International Conference on Machine Learning*, 2025.

Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi: 10.1126/science.220.4598.671.

Frédéric Lafitte, Jorge Nakahara Jr, and Dirk Van Heule. Applications of SAT solvers in cryptanalysis: finding weak keys and preimages. *Journal on Satisfiability, Boolean Modelling and Computation*, 9(1):1–25, 2014.

John Lazzaro, Sylvie Ryckebusch, Misha A. Mahowald, and Carver A. Mead. Winner-take-all networks of O(N) complexity. In *Advances in Neural Information Processing Systems*, pp. 703–711, 1988.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Zenan Li, Yunpeng Huang, Zhaoyu Li, Yuan Yao, Jingwei Xu, Taolue Chen, Xiaoxing Ma, and Jian Lu. Neuro-symbolic learning yielding logical constraints. In *Advances in Neural Information Processing Systems*, 2023.

Chuan Liu, Ruibing Song, Chunshu Wu, Pouya Haghi, and Tong Geng. InstaTrain: Adaptive training via ultra-fast natural annealing within dynamical systems. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=QhhShUQIpJ.

Chuan Liu, Chunshu Wu, Ruibing Song, Ang Li, Ying Nian Wu, and Tong Geng. An expressive and self-adaptive dynamical system for efficient function learning. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=KU7jlB2MPN.

Hao Lo, William Moy, Hanzhao Yu, Sachin Sapatnekar, and Chris H Kim. An Ising solver chip based on coupled ring oscillators with a 48-node all-to-all connected array architecture. *Nature Electronics*, 6(10):771–778, 2023.

Ying-Chang Lu, Chao-Ming Fu, Lien-Po Yu, Yen-Jui Chang, and Ching-Ray Chang. Quantum-inspired portfolio optimization in the QUBO framework. *arXiv preprint arXiv:2410.05932*, 2024.

Andrew Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2:5, 2014. doi: 10.3389/fphy.2014.00005.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Fabio Massacci and Laura Marraro. Logical cryptanalysis as a SAT problem. *Journal of Automated Reasoning*, 24(1):165–203, 2000.

Ilya Mironov and Lintao Zhang. Applications of SAT solvers to cryptanalysis of hash functions. In *International Conference on Theory and Applications of Satisfiability Testing*, pp. 102–115. Springer, 2006.

William Moy, Ibrahim Ahmed, Po-wei Chiu, John Moy, Sachin S Sapatnekar, and Chris H Kim. A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving. *Nature Electronics*, 5(5):310–317, 2022.

Carsten Peterson and Bo Söderberg. A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, 1(1):3–22, 1989. doi: 10.1142/S0129065789000414.

Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience*, 11:24, 2017. doi: 10.3389/fncom.2017.00024.

Kulin Shah, Nishanth Dikkala, Xin Wang, and Rina Panigrahy. Causal language modeling can elicit search and reasoning capabilities on logic puzzles. In *Advances in Neural Information Processing Systems*, 2024.

Anshujit Sharma, Richard Afoakwa, Zeljko Ignjatovic, and Michael Huang. Increasing Ising machine capacity with multi-chip architectures. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ISCA '22, pp. 508–521, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450386104. doi: 10.1145/3470496.3527414. URL https://doi.org/10.1145/3470496.3527414.

Ruibing Song, Chunshu Wu, Chuan Liu, Ang Li, Michael Huang, and Tony Tong Geng. DS-GL: Advancing graph learning via harnessing nature's power within scalable dynamical systems. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pp. 45–57, 2024. doi: 10.1109/ISCA59077.2024.00014.

Ruibing Song, Chuan Liu, Chunshu Wu, Ang Li, Dongfang Liu, Ying Nian Wu, and Tong Geng. DS-LLM: Leveraging dynamical systems to enhance both training and inference of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OPSpdc25IZ.

Weiwei Sun, Shengyu Feng, Shanda Li, and Yiming Yang. CO-Bench: Benchmarking language model agents in algorithm search for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33126–33134, 2026.

Kazuki Takabatake, Keisuke Yanagisawa, and Yutaka Akiyama. Solving generalized polyomino puzzles using the Ising model. *Entropy*, 24(3):354, 2022.

Sever Topan, David Rolnick, and Xujie Si. Techniques for symbol grounding with SATNet. In *Advances in Neural Information Processing Systems*, 2021.

Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*, 2020.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model, 2025. URL https://arxiv.org/abs/2506.21734.

Po-Wei Wang, Priya L. Donti, Bryan Wilder, and J. Zico Kolter. SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, 2019.

William Whitehead, Zachary Nelson, Kerem Y. Camsari, and Luke Theogarajan. CMOS-compatible Ising and Potts annealing using single-photon avalanche diodes. *Nature Electronics*, 6(12):1009–1019, 2023. doi: 10.1038/s41928-023-01065-0.

Logan G. Wright, Tatsuhiro Onodera, Martin M. Stein, Tianyu Wang, Darren T. Schachter, Zoey Hu, and Peter L. McMahon. Deep physical neural networks trained with backpropagation. *Nature*, 601:549–555, 2022. doi: 10.1038/s41586-021-04223-6.

Chunshu Wu, Ruibing Song, Chuan Liu, Yunan Yang, Ang Li, Michael Huang, and Tong Geng. Extending power of nature from binary to real-valued graph learning in real world. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=qT7DXUmX7j.

Zhun Yang, Adam Ishay, and Joohyung Lee. Learning to solve constraint satisfaction problems with recurrent transformer. In *International Conference on Learning Representations*, 2023.

A Background on DSM Hardware

A.1 Physical Implementation

A DSM computes by letting a physical system relax toward low-energy states. In an electronic DSM, each variable is a physical node, such as a capacitor voltage or an oscillator phase, and tunable coupling units, for example resistive or capacitive elements, connect pairs of nodes; local circuits add the field terms. Once the couplings and fields are set to those of a target energy, current flows between the nodes until the node states reach a low-energy configuration. A digital processor would reach the same configuration through a long sequence of arithmetic instructions; in a DSM it is the natural evolution of the circuit, which is the source of its latency and energy advantage on optimization-heavy workloads.

An Ising node compares its summed input with zero and settles into one of two states. A multi-state node of the kind used in DS-OPT (Figure 2b) keeps one input rail per state: the coupling units realize the full $D \times D$ table $\mathbf{J}_{ij}$, with a unit from rail w of node j to rail v of node i for every nonzero entry $J_{ij,vw}$ (same-state Potts couplings need only the $v \to v$ units), and add their contributions to each rail; the rails integrate these currents, and a winner-take-all circuit with injected noise selects the state with the largest integrated input, as in Eq. (4). For $D = 2$ the two rails reduce to a single comparator.

A.2 Continuous Quadratic DSMs Solve Linear Systems

AI-oriented DSMs such as NPGL (Wu et al., 2024) relax the binary state into a continuous one, $\mathbf{x} \in \mathbb{R}^N$ (we write $\mathbf{x}$, $\boldsymbol{\kappa}$ and $\mathbf{c}$ to avoid clashes with the symbols of Section 3), with a quadratic energy

$$\mathcal{H}_{\mathrm{cont}}(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^\top J\mathbf{x} + \frac{1}{2}\mathbf{x}^\top \operatorname{diag}(\boldsymbol{\kappa})\,\mathbf{x} - \mathbf{c}^\top\mathbf{x} = \frac{1}{2}\mathbf{x}^\top A_\kappa \mathbf{x} - \mathbf{c}^\top\mathbf{x}, \qquad A_\kappa = \operatorname{diag}(\boldsymbol{\kappa}) - J, \tag{8}$$

where J is the symmetric coupling matrix, $\boldsymbol{\kappa}$ contains the self-reaction coefficients of the nodes and $\mathbf{c}$ is an external drive. The ideal node dynamics are a gradient flow on this energy,

$$\frac{d\mathbf{x}}{dt} = J\mathbf{x} - \boldsymbol{\kappa} \odot \mathbf{x} + \mathbf{c} = -\nabla_{\mathbf{x}}\mathcal{H}_{\mathrm{cont}}, \qquad \frac{d\mathcal{H}_{\mathrm{cont}}}{dt} = -\left\|\nabla_{\mathbf{x}}\mathcal{H}_{\mathrm{cont}}\right\|_2^2 \le 0, \tag{9}$$

so the energy decreases until the state reaches an equilibrium. The circuit is stable when A_κ is positive definite. The energy is then convex with a single minimum, where $\nabla_{\mathbf{x}}\mathcal{H}_{\mathrm{cont}} = A_\kappa\mathbf{x} - \mathbf{c} = 0$, that is,

$$A_\kappa \mathbf{x}^\star = \mathbf{c}. \tag{10}$$

In its stable regime a continuous quadratic DSM therefore solves a linear system. This is efficient for real-valued transformations, but the landscape has no competing minima, so the combinatorial search of the discrete Ising energy, over 2^N configurations with many local minima, is lost. Recovering it requires higher-order interactions or nonlinear node circuits, which is the cost DS-OPT avoids by keeping the DSM discrete (Section 2).

B The Training Relaxation and Discrete Inference

B.1 Consistency of the Relaxed and Discrete Models

The relaxed energy is minimized at a discrete state. Extend Eq. (1) to states $\mathbf{m}_i$ in the probability simplex Δ^{D-1}. Because $\mathbf{J}_{ii} = 0$, the energy is linear in $\mathbf{m}_i$ when the other variables are fixed:

$$\mathcal{H}(\mathbf{m}) = -\,\mathbf{m}_i^\top \widetilde{\mathbf{f}}_i + \text{terms independent of } \mathbf{m}_i, \qquad \widetilde{\mathbf{f}}_i = \mathbf{h}_i + \sum_{j \ne i} \mathbf{J}_{ij}\mathbf{m}_j. \tag{11}$$

A linear function on the simplex is minimized at a vertex, the one-hot vector of the largest component of $\widetilde{\mathbf{f}}_i$. Starting from any minimizer over the product of simplices and replacing the variables by such vertices one at a time never increases the energy, so a one-hot state attains the relaxed minimum, and the minimum of the relaxed energy equals that of the discrete one. The relaxation thus changes the path to a low-energy state, not the states the energy prefers.

The relaxed update approaches the WTA at low temperature near steady state. As $T \to 0$, $\mathrm{softmax}(\mathbf{u}_i/T)$ tends to the one-hot vector of $\arg\max_v u_{i,v}$ (for a unique maximum), whereas the WTA of Eq. (4) selects the maximum of the instantaneous field $\mathbf{f}_i$. The two agree near the fixed point of the leaky integrator: with $\widetilde{\mathbf{f}}_i$ and r held fixed, Eq. (5) converges to $\mathbf{u}_i^\star = \eta\,\widetilde{\mathbf{f}}_i/(\lambda r)$, whose largest component is that of $\widetilde{\mathbf{f}}_i$. Away from this regime the relaxation is only a training surrogate, and Appendix B.2 checks empirically that the energies it trains are minimized by discrete annealing.

Gumbel-noise WTA samples the Boltzmann distribution. If $\xi_{i,v}$ are independent standard Gumbel variables, $\arg\max_v(f_{i,v} + T\xi_{i,v})$ equals v with probability $\mathrm{softmax}(\mathbf{f}_i/T)_v$ (the Gumbel-max identity (Maddison et al., 2014)), which is the conditional Boltzmann distribution of variable i given the others. The WTA update of Eq. (4) is therefore exact Gibbs sampling of the multi-state energy with Gumbel noise. We use Gaussian noise with the same variance, $\pi^2/6$, which analog circuits provide naturally; Gaussian noise only approximates the softmax, and in our experiments the two noise types differ in accuracy by less than half a point. Updating a block of coupled variables at once, as we do for image completion, is likewise an approximate parallel form of Gibbs sampling. For $D = 2$, with the encoding of Section 3.1, $f_{i,1} - f_{i,2} = 2\big(\sum_j J_{ij}\sigma_j + b_i\big)$, so

$$\arg\max_v \big(f_{i,v} + T\xi_{i,v}\big) = 1 \iff \sum_j J_{ij}\sigma_j + b_i + T\,\tfrac{\xi_{i,1}-\xi_{i,2}}{2} > 0, \tag{12}$$

which is the noisy comparator of an Ising machine.

B.2 Aligning Training with Discrete Inference

The argument above guarantees that the relaxed and discrete energies share their minimizers, but a network trained through a finite, hot relaxation need not produce an energy whose discrete minimum is the answer: if T_{final} stays high, the loss can be lowered by a soft state that no discrete state reproduces. The two measures of Section 3.3 remove this gap. Table 4 reports the accuracy under discrete inference of models trained with and without them.

Table 4: Accuracy (%) under discrete DSM inference (K=16, S=200) of models trained with different relaxations. Late cooling starts from the fixed value (Zebra $0.7 \to 0.1$, completion $0.5 \to 0.05$). Zebra: seed 0, 5k test puzzles; the $1.0 \to 0.1$ schedule of Table 1 gives 94.6 here and 94.4 on the full test set, and Table 1 reports the best of three seeds. Completion: full test sets. Separate training runs decoded with the Gumbel-noise WTA. —: not run.

Training relaxation	Zebra	Glyph 48	MNIST
Fixed T_{final} (0.7 Zebra, 0.5 completion)	81.0	78.4	73.9
+ late cooling	**94.3**	—	74.8
+ discrete forward pass	—	—	80.0
+ both	—	**87.5**	**80.2**

For Zebra, lowering T_{final} over the second half of training is enough. For image completion the discrete forward pass is what matters: with 2-state pixels and local couplings, the relaxed state can hold many pixels near $\sigma = 0$, a configuration that discrete spins cannot take. Running the forward pass with sampled discrete states, with the relaxation used only for the straight-through gradient, trains the network on the states it meets at inference, and late cooling adds a small further gain. Visual Sudoku needs neither measure: with the couplings fixed, a constant $T_{\mathrm{final}} = 0.7$ already gives discrete inference at the accuracy of Table 2.

C Experimental Details

C.1 Datasets

Zebra puzzles. We use the data of Shah et al. (2024): puzzles with 3–6 houses and 3–6 attributes, seven clue types and a unique solution. We train on 300k of the released 1.5M training puzzles and evaluate on the full released test set of 100k puzzles. A puzzle counts as solved only if every attribute value is placed in the right house. Attribute names are symbolic; the names in Figure 3a are for display only.

Visual Sudoku. We use the visual Sudoku data of Wang et al. (2019), in which every given digit of a board is replaced by an MNIST image, with the 9k/1k train/test split of Yang et al. (2023). Following the ungrounded protocol (Chang et al., 2020; Yang et al., 2023), no digit label is used: the loss covers only the blank cells. Which cells are given is known, as in all prior work. A board counts as solved only if all 81 cells are correct, including the given cells read from images, a stricter criterion than the blank-cell accuracy used by Yang et al. (2023).

Binary image completion. Glyph 24 and Glyph 48 are synthetic binary glyph images of 24×24 and 48×48 pixels with a masked square of 10×10 and 20×20 pixels; for MNIST we binarize the 28×28 digits and mask a 12×12 square. Each dataset uses 5k training images; the test sets have 200, 200 and 1k images. The metric is the pixel accuracy on the masked region; the all-background predictor of Table 3 is the trivial reference.

C.2 Models and Training

Table 5 lists the architectures and training settings. All models are trained with AdamW. The energy has the three forms of Section 3.2.

Zebra. The Transformer reads one token per clue, built from embeddings of the clue type and its items, together with one token per variable (a, v), the house of value v of attribute a, which gives up to 36 variables with $D \leq 6$ states; states beyond the number of houses are masked out. Each clue token also serves as one binary auxiliary DSM variable, so that three-way clues such as "in between" can be expressed with pairwise couplings; the network learns what the auxiliary variables mean. Couplings are constructed for all pairs except auxiliary–auxiliary pairs. The Transformer baseline uses the same encoder with a linear head over the houses.

Visual Sudoku. A LeNet-style CNN (LeCun et al., 1998) with two convolutional and two linear layers (1.20M parameters) maps each given cell image to digit probabilities $\mathbf{p}_i$, which set the field $\mathbf{h}_i = \tau \log \mathbf{p}_i$; blank cells receive no field. The shared couplings of Eq. (6) are learned beforehand from symbolic Sudoku alone, on 1k Sudoku-Extreme training puzzles without augmentation, by maximizing the pseudo-likelihood of the solutions (Besag, 1975) with an ℓ_1 penalty on J; no digit image or label is involved. They are then kept fixed while the LeNet, trained from scratch, and τ are trained through the DSM relaxation on the blank-cell loss. The couplings are thus the one component of our models that is not trained through the DSM relaxation, and they are trained on different data from the rest of the model. Pseudo-likelihood conditions each cell on the true digits of all other cells; symbolic puzzles provide these digits, but image boards without digit labels do not, which is why the LeNet cannot be trained this way.

Image completion. The Transformer reads pixel tokens (2×2 patches for Glyph 48). Each pixel is a binary DSM variable; the 27 coupling displacements are drawn once, with a fixed seed, from a half-plane of the 17×17 window, so each pixel interacts with up to 54 others. Observed pixels are clamped. The Transformer baseline uses the same encoder with an MLP head whose width matches the parameter count.

Table 5: Architectures and training settings. K_r, λ, η, T_init and T_final are the relaxation parameters of Eq. (5); "late cooling" ($a \to b$) lowers T_final geometrically over the second half of training. Aux.: binary auxiliary variables.

	Zebra	Visual Sudoku	Image completion
Neural layers	Transf., d=128, 6 layers	CNN (2 conv, 2 FC)	Transf., d=128, 4 layers
Couplings	pair MLP	shared, d_q=64	27 displacements
DSM variables	≤ 36 ($D \leq 6$), ≤ 40 aux.	81 (D=9)	576/2304/784 (D=2)
K_r, λ, η, T_init	30, 0.1, 0.5, 2.0	30, 0.1, 0.5, 2.0	24, 0.1, 0.5, 2.0
T_final	$1.0 \to 0.1$	0.7	$0.5 \to 0.05$
Discrete forward	no	no	yes
Epochs, batch, lr	30, 128, $5{\times}10^{-4}$	50, 64, 10^{-3}	100, 32, $3{\times}10^{-4}$
Training data	300k puzzles	9k boards	5k images

C.3 Annealing Budgets and DSM Sizes

All results use the same annealing budget: $K = 16$ independent anneals per instance, each of $S = 200$ sweeps, with the noise amplitude T_t lowered geometrically from 2.0 to 0.05 (0.02 for visual Sudoku); the lowest-energy state of the K anneals is returned. Zebra and visual Sudoku update one variable at a time in random order; image completion updates random blocks of 25% of the pixels at once, an approximate parallel form of the sequential update. Table 6 lists the DSM size of each task.

Table 6: DSM size per instance and annealing budget. Free variables are those the DSM sets: given Sudoku cells receive a field but stay free, so misreadings can be corrected, while clamped pixels only contribute fields. Zebra: average and maximum over the test set (at most 36 task and 40 auxiliary variables).

Task	Free variables	States D	Partners per variable	$K \times S$
Zebra	34.8 avg. (max. 71)	≤ 6; aux. 2	all active	16×200
Visual Sudoku	81	9	80 (shared)	16×200
Glyph 24 / 48, MNIST	100 / 400 / 144	2	≤ 54	16×200

C.4 Timing, Energy and Parameter Counts

GPU time. All training and inference times are measured on single NVIDIA A100 GPUs. Inference times cover the full test set, in fp32, with the batch sizes of our evaluation, after a warm-up pass and with CUDA synchronization, taking the best of three runs. For DS-OPT, the GPU part is the construction of the energy, the neural layers and the energy projections. Two implementation details make it efficient without changing its output: in Zebra, the pair MLP is evaluated only on the pairs that can carry a coupling (both tokens present, not both auxiliary), and in image completion the coupling head, which is linear in the concatenated features, is evaluated as two per-pixel projections followed by one gather over all displacements. The Transformer baselines are timed with the same script, and those of the Recurrent Transformer with its released code on the same GPU type.

DSM time and energy. The DSM part, the per-instance normalization of the energy and the annealing with the budgets of Table 6, is evaluated by circuit-level simulation with Cadence Virtuoso in a 45 nm process, over the full test set. We have also run the 2-state form of DS-OPT on an existing coupled-oscillator Ising chip (Cılasun et al., 2025), where its behavior agreed well with the simulation. The total inference time of DS-OPT is the sum of the GPU and DSM times.

Energy. GPU energy is inference time multiplied by 250 W per A100. The inference energy of DS-OPT is its GPU energy plus the simulated DSM energy.

Baselines from the literature. For Shah et al. (2024), whose code is written in JAX, we time an architecture-equivalent PyTorch model (8 layers, width 576, 42M parameters) on the same GPU: inference is greedy decoding of the answer tokens with a key–value cache, and training time is the measured step time multiplied by the number of steps in their configuration. The Recurrent Transformer's training time is its measured step time multiplied by its default schedule.

Parameters. We count the learned parameters of the network. For DS-OPT, the fields and couplings are recomputed for every input and are not parameters; the shared Sudoku couplings are counted through the parameters that produce them. For recurrent models we also give the count of the equivalent unrolled network (Table 2).

D Mapping a Multi-State Energy to a Binary Ising Machine

A multi-state energy can also run on binary Ising hardware. Represent variable i by D binary spins $q_{i,v} \in \{0, 1\}$, the entries of $\mathbf{s}_i$, and add a penalty that is zero exactly on one-hot states:

$$\mathcal{H}_{\text{bin}}(\mathbf{q}) = -\sum_i \mathbf{h}_i^\top \mathbf{q}_i - \frac{1}{2}\sum_{i \neq j} \mathbf{q}_i^\top \mathbf{J}_{ij} \mathbf{q}_j + \mu \sum_i \Big(\sum_v q_{i,v} - 1 \Big)^2 . \tag{13}$$

On one-hot states $\mathcal{H}_{\text{bin}}$ equals Eq. (1). Let $F = \max_{i,v} \left(|h_{i,v}| + \sum_{j \neq i} \sum_c |J_{ij,vc}| \right)$ bound the magnitude of every local field $f_{i,v}$. If $\mu > F$, every state that is not one-hot has a single-spin flip that lowers the energy: activating a spin of a variable with no active spin changes the energy by $-(f_{i,v} + \mu) < 0$, and deactivating one of $n \geq 2$ active spins changes it by $f_{i,v} - \mu(2n - 3) \leq f_{i,v} - \mu < 0$. All local minima of Eq. (13), and hence its global minimum, are then one-hot, and the binary machine returns an assignment of the multi-state problem. With $q_{i,v} = (1 + \sigma_{i,v})/2$ the energy takes, up to an additive constant, the Ising form of Eq. (3) over ND spins $\sigma_{i,v} \in \{-1, +1\}$.

Two steps make the mapping easier to anneal. First, the couplings can be gauged without changing the relative energies of one-hot states, and hence the minimizer: subtracting the row and column means of each table $\mathbf{J}_{ij}$ and adding the corresponding terms to the fields changes one-hot energies only by a constant, because $\mathbf{1}^\top \mathbf{q}_j = 1$ on those states, and centering each field $\mathbf{h}_i$ does the same. Gauging shrinks the local fields, and with them the penalty that F requires. Second, the penalty can be ramped up during the anneal, so that a variable can change its value through states that are briefly not one-hot early on, while the final low-noise sweeps enforce the encoding. The price of the mapping is the larger number of spins, ND instead of N, and the penalty couplings; the multi-state DSM of DS-OPT avoids both.