POLICY RBMS ON SUPERCONDUCTING QUANTUM COMPUTERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Restricted Boltzmann machines (RBMs) are Ising models, and a quantum processor that holds the Gibbs state of an Ising model returns a joint sample of all visible and hidden units of the RBM in a single measurement, without Markov chains or burn-in. Two obstacles stand in the way: the Boltzmann factor is not unitary, so the state is hard to prepare, and superconducting qubits couple to few neighbours, so an RBM whose graph does not match the hardware costs circuit depth that the quantum processor does not have. We treat both as a representation-learning problem in which the quantum processor defines constraints on the model space. A neural network policy for power-grid control, trained by AlphaZero-style self-play, is distilled into a conditional RBM whose 118 units and 210 weights occupy every functioning qubit and coupler of a 120-qubit IBM Nighthawk r2 quantum processor, whose action code follows the measured qubit noise and whose binary input state code is learned with the model. A quantum imaginary-time evolution prepares its conditional Gibbs state. A series of experiments isolates the effect of each technique. The quantum-processor policy agrees with the teacher's preferred action in about half of the held-out states, against 27 % for the state-independent prior, and lies within a few hundredths of a nat of its exact read-out, whereas QAOA-type circuits fall far short of it. To our knowledge this is the largest RBM sampling experiment so far on a gate-based superconducting quantum processor with a read-out that recovers the trained model.

1 INTRODUCTION

Restricted Boltzmann machines (RBMs) (Smolensky, 1986; Hinton, 2002; Ackley et al., 1985) sit between machine learning and quantum computing: learning models over binary units, and at the same time classical Ising models whose Gibbs distribution is the thermal distribution of a Hamiltonian with pairwise couplings on the edges of the bipartite graph. A superconducting quantum processor is a lattice of qubits with couplers on the lattice edges, and preparing and measuring a thermal-like state of an Ising Hamiltonian on that lattice is one of the few things such a quantum processor can attempt at full scale today. RBMs are therefore a natural meeting point. Prior work explored it mainly on quantum annealers (Dumoulin et al., 2014; Benedetti et al., 2016; 2017; Amin et al., 2018), which embed fully connected RBMs of up to $120 + 120$ units by chaining physical qubits ($256 + 256$ with unit dropout) and sample at an effective temperature that has to be estimated (Adachi & Henderson, 2015; Job & Adachi, 2020; Caldeira et al., 2020; Dixit et al., 2021; 2022; Kālis et al., 2023; Salmenperä & Nurminen, 2023; Moro & Prati, 2023; Higham & Bedford, 2023; Sinno et al., 2025; Salloum et al., 2025; El Yazizi et al., 2025; El-Yazizi & Koshka, 2025). Appendix G surveys these experiments. On gate-based quantum processors there are proposals (Kieferová & Wiebe, 2017; Verdon et al., 2017; Anschuetz & Cao, 2019; Zoufal et al., 2021; Coopmans & Benedetti, 2024) and experiments on a few qubits (Zhu et al., 2020; Wu et al., 2020; Sagastizabal et al., 2021; Layden et al., 2023). The annealer experiments use benchmark images, synthetic patterns or compressed features, the gate-based ones thermal states of physical models, and none samples the decision policy of a control task. We show that such a policy fits a gate-based quantum processor.

What the quantum processor offers is the sample itself. The joint distribution of an RBM over its visible and hidden units is the Gibbs state $\rho \propto e^{-\hat{E}}$ of its Ising Hamiltonian, and a quantum processor that holds this state returns a joint sample of all visible and hidden units by one measurement of

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.

its qubits: no Markov chain, no burn-in and no convergence diagnostics. In the long term, quantum processors can thus replace Markov chain Monte Carlo as samplers without inheriting its drawbacks. The difficulty is the preparation: $e^{-\hat{E}}$ is not unitary and cannot be applied as a gate sequence, and the known routes to it carry numerical and complexity problems (Section 4.1).

The second obstacle is connectivity. A superconducting quantum processor couples each qubit to only a few neighbours, because every additional coupler adds crosstalk and frequency collisions and complicates control (Krantz et al., 2019; Chamberland et al., 2020; Hertzberg et al., 2021). Two qubits that are not neighbours can still be entangled, by routing one of them through the lattice with SWAP gates until the two logical qubits sit on adjacent physical ones (Zulehner et al., 2019; Li et al., 2019; Cowtan et al., 2019), but every SWAP costs three two-qubit gates and adds to the circuit depth, and depth is what gate errors and decoherence limit on today's quantum hardware (Preskill, 2018). Unless the graph of an RBM is a subgraph of the lattice, the depth of any circuit that realises its couplings therefore grows with the SWAPs its edges require. Which topology of the RBM and which binary encoding of its inputs and outputs represent the task well and fit the quantum processor at the same time is the subject of this paper. Quantum annealers face the same constraint, with chains of physical qubits in place of routing and a failure rate that grows with the chain length, so our methods transfer to annealers as well. Appendix G explains the correspondence.

We approach both obstacles through *knowledge distillation*. Model compression (Bucilă et al., 2006) and its later form, knowledge distillation (Hinton et al., 2015), train a small *student* to reproduce the outputs of a large *teacher* on the teacher's own inputs, using the teacher's soft output distribution rather than hard labels as the training signal, so that the student inherits the teacher's generalisation without its size. For a control policy this is policy distillation (Rusu et al., 2016). Our teacher is a deep neural network policy for power-grid topology control, trained by AlphaZero-style self-play with tree search (Silver et al., 2017) on the `l2rpn_case14_sandbox` environment of the Learning-to-Run-a-Power-Network challenge (Marot et al., 2020; 2021): in critical states of a 14-substation grid with 20 lines it chooses among 82 busbar reconfigurations, and its decision log provides its action distribution for 4551 decisions. The student is a conditional RBM whose visible layer carries binary codes of state and action, with at most 120 units and, finally, only couplers of IBM's Nighthawk lattice (IBM Quantum, 2026) as connections. Any teacher with a finite action set and any Boltzmann machine whose graph maps onto the lattice can take their place. We chose the model so that every result of the 118-qubit system can be validated against its classically available exact read-out. We make no claim that the quantum processor accelerates the RBM, but study which steps bring a classical RBM onto a quantum processing unit (QPU) and what each is worth. The steps can and should be transferred to probabilistic models whose inference is classically intractable, where the same lattice, codes and preparation apply and only the check falls away.

Our main result is a QPU-aware representation of the distilled policy. We analyse the topology of the quantum processor and the noise of its qubits and let both define constraints on the model space: the graph of the RBM is the lattice of `ibm_phoenix` with all 118 functioning qubits and all 210 functioning couplers, its action code places the actions on strings that the measured flip rates leave intact, and its input state code is learned for that register. A classically compiled imaginary-time evolution prepares the conditional Gibbs state of this model, and the policy read from the quantum-processor samples agrees with the teacher's preferred action in about half of the held-out states, against 27 % for the state-independent prior. A series of experiments isolates the effect of each of these techniques. To our knowledge no Boltzmann machine of this size has been sampled on a gate-based quantum processor before, and what stood in the way was not the qubit count but the two obstacles above together with a third that only the experiments revealed, each removed by a reusable piece of our recipe: a model graph that fits the lattice without routing, an output code that a sampler can read, and a Gibbs-state preparation that is exact at shallow depth. These are our three contributions: (1) *A distillation pipeline for lattice-shaped RBMs* (Section 3): a training-free closed form, fine-tuning with a jointly learned binary state encoding, a fan-in penalty with projection, and a structure-aware closed form trained on the qubit lattice. (2) *Binary codes as the interface between task and sampler* (Section 3.6): a dense random action code gives the strongest exact model but is unreadable from samples, whereas a fully assigned register is readable at a cost in capacity that noise-aware strings, sharpened targets and a register-specific state code recover. (3) *Two families of Gibbs-state preparations, compared on quantum hardware* (Section 4): QAOA-type circuits, from the single pseudo-Boltzmann layer of Díez-Valle et al. (2023) to five layers with angles tuned to the model, and a classically compiled quantum imaginary-time evolution (Motta et al., 2020).

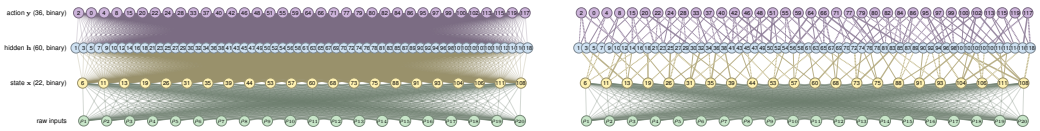

Figure 1: Left: a dense RBM with the unit counts of the grid model, 22 state, 60 hidden and 36 action units, and all visible–hidden edges. Right: the sparse grid RBM with 210 edges in the same layered form. Colours: state units yellow, hidden units blue, action units lavender, raw inputs (the 20 continuous line loadings) green. Edge colour marks the visible end. On the right, line width shows the magnitude of the learned weight (saturating at $|W| = 6$) and a light/dark pattern a negative weight. The raw inputs feed every state unit through the threshold layer (4), which learns the state code and stays dense in every model because it is evaluated classically. The three layers above are binary.

2 Background

This section fixes the view of an RBM as an Ising model, the circuits that approximate its Gibbs state, the quantum processor and the task.

Boltzmann machines as Ising models A restricted Boltzmann machine (Smolensky, 1986; Hinton, 2002; Fischer & Igel, 2014) is a bipartite Markov random field over binary visible units v and hidden units h with energy $E(v, h) = -v^{\mathsf{T}}Wh - b^{\mathsf{T}}v - c^{\mathsf{T}}h$ and Gibbs distribution $p(v, h) \propto e^{-E(v,h)}$. With spins $\sigma = 2v - 1$ the energy is a classical Ising Hamiltonian on the bipartite graph. Our RBM is *conditional* (Taylor et al., 2007; Larochelle & Bengio, 2008; Larochelle et al., 2012): the visible layer holds binary codes of a state and of an action, and at use time the state units are *clamped*, i.e. fixed to the code of the observed state while only action and hidden units are free, so the object of interest is the conditional Gibbs distribution of the action and hidden units given the state.

Gibbs states on quantum computers Preparing a Gibbs state on a quantum processor is the central difficulty (Temme et al., 2011; Brandão & Kastoryano, 2019; Wu & Hsieh, 2019) (Section 4.1). We compare alternative approaches: circuits of the quantum approximate optimisation algorithm (QAOA) type (Farhi et al., 2014), of which one layer already produces a pseudo-Boltzmann state (Díez-Valle et al., 2023), and quantum imaginary-time evolution (QITE) (Motta et al., 2020), which replaces every non-unitary step by a unitary on a domain of qubits around the term.

The quantum processor IBM's Nighthawk QPUs (IBM Quantum, 2026) have 120 fixed-frequency qubits on a 12×10 square lattice with 218 tunable couplers. The r2 quantum processor `ibm_phoenix` had two defective qubits, 17 and 62, at the time of the experiments (Fig. 2, right), whose removal leaves 118 qubits and 210 couplers, exactly the grid RBM.

Problem and evaluation The teacher's log holds $N = 4551$ decisions from 100 scenarios, each a pair (ρ_i, π_i) of the 20 line loadings $\rho_i \in [0, \infty)^{20}$ and the teacher's action distribution $\pi_i \in \Delta^{81}$. The teacher is intrinsically stochastic: its mean entropy on the held-out rows is 3.02 nats. We hold out 20 % of the scenarios (all decisions of a scenario on the same side), 816 validation rows for the split used throughout. The student defines $q(a \mid s)$ over the 82 actions. We report the conditional cross-entropy of the teacher under the student, $\mathrm{CE} = -N_{\mathrm{val}}^{-1} \sum_i \sum_a \pi_i(a) \log q(a \mid s_i)$, bounded below by the teacher's entropy 3.02, and the *top-k agreement*, the fraction of rows in which the teacher's arg-max action is among the k most probable actions of the student. The state-independent mean policy $\bar{\pi}$ scores $\mathrm{CE} = 3.781$ and top-1/2/3 $= 0.270/0.479/0.569$.

3 Learning the RBM

This section builds the model in stages, from the conditional RBM and its codes to the action registers that make it sampleable. Fig. 1 shows the dense model (left) and the sparse grid model in layered form (right), Fig. 2 the latter on the lattice.

3.1 THE CONDITIONAL RBM

The model has three groups of binary vertices: n_s *state* (input) units $\mathbf{x}$, n_a *action* (target) units $\mathbf{y}$ and n_h hidden units $\mathbf{h}$, with $n_s + n_a + n_h \le 118$ on the lattice. Visible $(\mathbf{x}, \mathbf{y})$ and hidden units form a bipartite graph $\mathcal{E}$. The energy is $E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = -(\eta^\top \mathbf{x} + \xi^\top \mathbf{y} + \sum_h b_h h_h + \sum_{(v,h)\in\mathcal{E}} W_{vh}\, v\, h_h)$, where v ranges over visible units, and summing out the hidden units gives the free energy

$$F(\mathbf{x}, \mathbf{y}) = \eta^\top \mathbf{x} + \xi^\top \mathbf{y} + \sum_h \operatorname{softplus}\big(b_h + W_{\cdot h}^\top (\mathbf{x}, \mathbf{y})\big). \tag{1}$$

Each action a has a fixed binary code $c(a) \in \{0,1\}^{n_a}$, and the model's policy is the softmax of the free energy over the admissible codes,

$$q(a \mid s) = \frac{\exp F(\mathbf{x}(s), c(a))}{\sum_{a'=1}^{82} \exp F(\mathbf{x}(s), c(a'))}. \tag{2}$$

The normaliser runs over 82 codes only, never over the state units, so the distillation objective

$$L(\theta) = -\frac{1}{\sum_{i,a} w_{ia}} \sum_i \sum_a w_{ia}\, \log q(a \mid s_i), \qquad w_{ia} = \pi_i(a), \tag{3}$$

is exact and needs neither sampling nor a partition function. The teacher's soft target π_i enters as one weighted row per (state, action) pair, so the model matches the whole distribution rather than the arg-max. Rows sharing a state are grouped, so an evaluation costs $O(S \cdot 82 \cdot (|\mathcal{E}| + n_h))$ for S distinct states. It is the conditional counterpart of the classification RBM (Larochelle et al., 2012). A hidden unit without any edge to an action unit cancels in (2) and is *dead* for the read-out.

The *action code* is an output code in the sense of error-correcting output codes (Dietterich & Bakiri, 1995): the 7-bit binary *index* of the action, a *two-hot* code in n_a bits (each action lights two bits), or a dense random code. Every *state code* is a layer of n_s linear threshold units on the standardised loadings $\tilde{\rho} = (\rho - \mu)/\sigma$,

$$x_j = [\, V_j^\top \tilde{\rho} + c_j \ge 0 \,], \quad j = 1, \dots, n_s. \tag{4}$$

The hand-made "one-bit" code, which sets bit j when line j is loaded above 0.8, is one instance. The *learned* code of Section 3.3 trains (V, c) jointly with the RBM. The layer is evaluated classically and stays dense in every model, because the state bits are clamped before anything is sampled.

3.2 CLOSED-FORM INITIALISATION

Before any gradient step, we write the RBM directly from the data by a Parzen-type construction that ties one hidden unit to one visible configuration $\mathbf{z}^{(h)} \in \{0,1\}^{n_s+n_a}$ with mass p_h and writes

$$W_{vh} = \alpha_v\, (2 z_v^{(h)} - 1), \qquad b_h = C + \log p_h - \sum_{v \in N(h)} \alpha_v z_v^{(h)}, \tag{5}$$

with $\eta = \xi = 0$, where $N(h)$ is the set of visible units connected to h. Each hidden unit then measures an α-weighted Hamming distance to its prototype, and the visible marginal is a kernel density estimate with the exponential Hamming kernel (Parzen, 1962). With one hidden unit per distinct configuration it attains the maximum-likelihood estimate for large α and C (Appendix A.1). With $n_h \approx 90$ units for several thousand distinct configurations the prototypes must be selected greedily (Nemhauser et al., 1978): our *cegreedy* rule adds the configuration whose closed-form unit most reduces the training cross-entropy (3) and finally merges every configuration's mass into its nearest prototype. Simulated annealing (Kirkpatrick et al., 1983) finds no better set (Appendix A).

3.3 FINE-TUNING AND LEARNED BINARISATION

The closed form is a good initialisation but not a stationary point of (3). Since the objective is smooth we minimise it directly from the closed-form weights, with the limited-memory BFGS method (L-BFGS, Liu & Nocedal, 1989) for dense models and the fast iterative shrinkage-thresholding algorithm (FISTA, Beck & Teboulle, 2009) with backtracking whenever a proximal term is present.

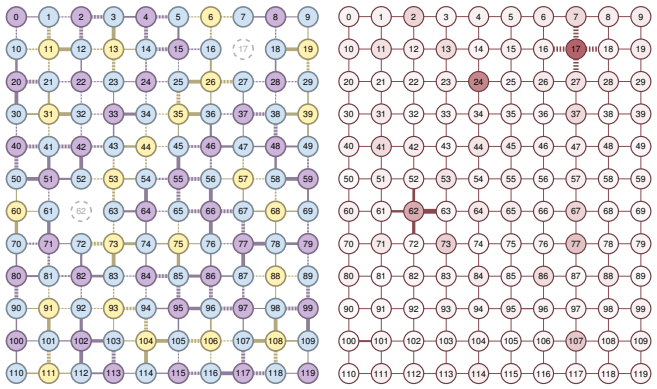

Figure 2: Left: the grid RBM of Fig. 1 on the 12×10 lattice (vertices numbered row by row from 0, top left, to 119, the defect vertices 17 and 62 removed and dashed). Colours, edge colour, width and pattern are as in Fig. 1. Right: calibration of `ibm_phoenix` in the same layout. The vertex fill shows the read-out error from white (0) to dark red (≥ 0.10) and the edge width the controlled-Z (CZ) error, with couplers at error 1.0 dashed. The medians are 0.008 and 0.002, and qubits 17 and 62 lie far above.

Fine-tuning the one-bit closed form lowers the cross-entropy by a third of a nat, and it changes what a code is worth, so the code is learned together with the model.

We train the state code *alternating* with the RBM in the pattern of an expectation-maximisation (EM) loop (Dempster et al., 1977): in each round the rows are re-encoded, the sufficient statistics are refreshed, the RBM weights are re-fitted, and then the threshold units take a fixed number of optimiser steps with the weights fixed. In *hard* rounds the bits are binary in the forward pass and the step function is back-propagated through a sigmoid surrogate (straight-through estimator, Bengio et al., 2013; Hubara et al., 2016), using the RBM's exact gradient $\partial L/\partial \mathbf{x}$. In *soft* rounds the bits are $\sigma(\kappa z)$ in the forward pass as well and both sides minimise the relaxed objective, with the slope κ annealed from 4 to 30. The objective depends on the rows only through the weights of the (state, string) pairs, so it is evaluated on that histogram at a cost independent of the number of rows. Standardised features were essential, and the start and the schedule change the result by a few hundredths of a nat (Appendix B).

3.4 Fan-In-Limited RBM

A hidden unit with a bounded number of inputs motivates the constraint that every hidden unit has at most K non-zero incoming weights. The trimmed ℓ_1 penalty $P_K(W) = \sum_h \sum_{v \notin \mathrm{top}_K(h)} |W_{vh}|$ has a proximal operator (Parikh & Boyd, 2014) that soft-thresholds exactly the entries outside each unit's top-K. After dense pre-training, proximal runs with λ growing geometrically are followed by a hard projection onto each unit's top-K and a masked fit. Two findings from the fan-in study in Appendix C shape the sparse models: the two-hot action code is the right code for small fan-in, since a unit reading four bits of a 7-bit index cannot identify an action, and re-training the state code against the projected model is the single most effective step. Both hold for a pruned dense model. Trained on the lattice from the start, the model prefers the dense random code (Appendix D).

3.5 Grid-Embedded RBM

The fan-in models prune a dense RBM. The final model fixes the structure up front: the RBM is embedded into the 12×10 grid of Fig. 2 with the defect vertices 17 and 62 removed, 118 vertices and 210 edges. A grid graph is bipartite: one colour class becomes the hidden layer, the other the visible layer, the grid edges are the only RBM edges, and every hidden unit sees its 2 to 4 neighbours. The visible class carries the action code and learned state bits. Since a hidden unit whose neighbours are all state bits cancels in (2), a local search places the action bits such that every hidden unit has a neighbour of each kind. Training under the topology of a physical quantum processor is the problem of Dumoulin et al. (2014) and Benedetti et al. (2017). Here two changes make the closed form structure-aware: the bias in (5) sums only over the connected visible units $N(h)$, so a unit is a matched filter on its neighbourhood, and we select prototypes *per unit*, greedily adding the (unit, local pattern) pair that most reduces the training cross-entropy.

3.6 Action Registers

A dense random code leaves most strings without an action, so its samples must be decoded to the nearest code. The alternative is an action register of k bits whose 2^k strings are *all* assigned to actions: strings 0 to 81 are the actions themselves, the remaining $2^k - 82$ strings go to the most frequent actions in proportion to their prior mass, and an action with m_a strings enters training with weight $\pi(a)/m_a$ on each of them, so the model splits its mass evenly over its strings and the read-out sums them back. The cost on the grid is capacity: k action vertices cover at most $4k$ hidden units, so with $k = 8$ almost half of the hidden units are dead for the read-out.

Three measures recover the cost. The *noise-aware string assignment* places the base string of every action greedily, in prior order, at maximum Hamming distance from the strings already placed, with bit i weighted by $-\log f_i$ for the flip probability f_i of its qubit (Section 4.4), and fills each action's extra strings with its nearest unassigned strings, so that flips of unreliable bits stay inside an action. *Sharpened targets* replace the training target by $\pi^2/\sum_a \pi^2$, so that the objective rewards the teacher's preferred action more than its tail (the metrics stay on π), a deliberate departure from matching the whole distribution for a model read out through its arg-max, and at a fixed state code it costs no cross-entropy. A *register-specific state code* is transferred from the learned 10-bit register or learned for the wide register itself, which the histogram form of the encoder objective makes affordable.

4 Quantum State Preparation

This section moves the grid RBM onto the quantum processor: its Ising form, two families of preparations of its conditional Gibbs state, and the read-out.

4.1 Gibbs States and the Ising Form on the Lattice

The Gibbs state of a Hamiltonian $\hat{H}$ at inverse temperature β is the density operator

$$\rho_\beta = e^{-\beta\hat{H}} \big/ \operatorname{Tr} e^{-\beta\hat{H}}. \tag{6}$$

For the classical, diagonal Hamiltonian of an RBM it is diagonal in the computational basis with entries $p_\beta(\sigma) = e^{-\beta\hat{E}(\sigma)}/Z$, so measuring every qubit of a quantum processor that holds ρ_β, or the coherent Gibbs state $|\psi_\beta\rangle \propto \sum_\sigma \sqrt{p_\beta(\sigma)}|\sigma\rangle = e^{-\beta\hat{E}/2}|+\rangle^{\otimes n}$ with the same statistics, returns one joint sample of all visible and hidden units. At $\beta = 1$ these statistics are the RBM's joint distribution itself. The difficulty is that $e^{-\beta\hat{E}/2}$ is not unitary and cannot be applied as a gate sequence. Trotterisation, the standard route to real-time evolution (Lloyd, 1996; Suzuki, 1976; Childs et al., 2021), splits imaginary time into small steps as well, but each step $e^{-\Delta\tau h_j}$ must then be replaced by a unitary fitted from measurements of the current state through a linear system that grows exponentially with the domain and is ill-conditioned in practice (Motta et al., 2020; Nishi et al., 2021), and ancilla-based steps, quantum Metropolis sampling and block encodings need post-selection or depths beyond present quantum hardware (Temme et al., 2011; Chowdhury & Somma, 2017; Rall et al., 2023; Chen et al., 2023). For a classical Hamiltonian, however, the fitted unitaries can be computed without any measurement, which Section 4.3 exploits.

The RBM of Fig. 2 maps onto `ibm_phoenix` with vertex number = physical qubit number, every RBM edge a lattice coupler and the two defective qubits never touched. With spins $\sigma_i = 2v_i - 1 \in \{\pm 1\}$ for every unit, the RBM energy is a classical Ising Hamiltonian on the lattice,

$$\hat{E}(\sigma) = \sum_{(v,h)\in\mathcal{E}} J_{vh}\,\sigma_v\sigma_h + \sum_i h_i\,\sigma_i, \qquad J_{vh} = -\tfrac{1}{4}W_{vh}, \tag{7}$$

with fields h_i from the biases and weight sums. The RBM's joint distribution is its Gibbs state at inverse temperature one.

4.2 QAOA-Based Gibbs-State Approximation

The first preparation is the single-layer circuit of Díez-Valle et al. (2023), $|\Psi(\gamma,\theta,\lambda)\rangle = R_y(\theta)^{\otimes N}\,U_1(\lambda)^{\otimes N}\,e^{-i\gamma\hat{E}}\,H^{\otimes N}|0\rangle^{\otimes N}$ with the phase gate $U_1(\lambda) = \operatorname{diag}(1, e^{i\lambda})$, a generalisation

of one QAOA layer (Farhi et al., 2014), whose measurement probabilities follow a Boltzmann law in the classical energy up to fluctuations, at an effective inverse temperature that grows with γ up to the angle that minimises $\langle \hat{E} \rangle$. As in the parameter-setting strategy of Streif & Leib (2020) we fix the angles classically, by minimising $\langle \Psi | \hat{E} | \Psi \rangle$ exactly over the light cones of the single layer, so the quantum processor is used only to sample. Matching the optimal energy to the mean energy of the classical Gibbs distribution (Benedetti et al., 2016) gives an effective inverse temperature of about 0.2: one layer samples a version of the RBM that is four to five times hotter than the model itself. For the register models the conditional components are small stars, so we also emulate the circuit exactly and tune its angles to the model by a Kullback–Leibler objective on the training states, for one and for p layers (Appendix E).

4.3 Imaginary-Time Evolution

Since all terms of $\hat{E}$ commute, imaginary-time evolution towards the coherent Gibbs state of (6) has no Trotter error. Variational QITE (McArdle et al., 2019) would need $O(P^2)$ circuits per step for an ansatz with P parameters. For a circuit as expressive as our models, with 328 (grid RBM: 210 couplings, 118 fields) to about 3000 (dense RBM) Ising parameters, that means 10^5 to 10^7 circuits per step. We use instead the Trotterised QITE of Motta et al. (2020), which replaces every local non-unitary step $e^{-\Delta\tau h_j}$ by a local unitary $e^{-i\Delta\tau A_j}$ on a domain around the term, with coefficients that solve a small linear system of expectation values in the current state. Normally these require measurements on the quantum state. For a classical Hamiltonian they do not, because the state after k exact steps is the coherent Gibbs state at $\beta = 2k\Delta\tau$, so every entry is a Gibbs expectation that we estimate by Metropolis sampling (Metropolis et al., 1953; Hastings, 1970) of the conditional model. The compiled circuit is a fixed sequence of two-qubit unitaries on the lattice edges and single-qubit rotations, without measurement feedback or variational loop.

The *edge-local* variant uses the two qubits of a coupling as the domain: its unitary cannot reproduce the effect of $e^{-\Delta\tau h_j}$ on correlations that extend beyond them, so its error shrinks only with $\Delta\tau$ and the depth grows with the number of steps, but it applies to any model, including the 96-qubit component of the 36-bit grid RBM. The *QITE limit* takes the domain to be a whole connected component of the coupling graph among the free qubits. In this limit of QITE the update is exact for any $\Delta\tau$, and one step maps $|+\rangle^{\otimes n}$ onto the coherent Gibbs state of the component. For the register models this is inexpensive, because their free-qubit coupling graph is a set of stars: every action qubit has up to four private hidden neighbours, and the remaining hidden qubits are dead singletons. For a star the coherent Gibbs state factorises as $\sqrt{p(\sigma_c)} \prod_l \sqrt{p(\sigma_l \mid \sigma_c)}$ over centre c and leaves l, so the exact update is one R_y on the centre and one uniformly controlled R_y (two controlled-NOT gates) per leaf. Appendix F draws the circuit. The same construction covers every tree component, with the root marginal and the parent-child conditionals from the sum-product messages of the tree (Pearl, 1988; Kschischang et al., 2001) and one uniformly controlled R_y (Möttönen et al., 2005) per edge (Grover & Rudolph, 2002; Low et al., 2014). The QITE-limit step is shallow and exact but applies only when the components are trees or small, i.e. for the register models, whose conditional components are stars (in the 16-bit model one tree links three register bits, which the same tree construction prepares).

4.4 Conditional Sampling, Read-Out and Mitigation

We prepare the state qubits deterministically, applying X for bit 0 and nothing for bit 1, since $|0\rangle$ encodes $\sigma = +1$. The coupling term of such a qubit reduces exactly to a single-qubit rotation on its neighbour, so only the hidden–action couplings need two-qubit gates, and the layout is the identity. We measure the free qubits, i.e. one joint sample of all action and hidden units given the clamped state, discard the hidden bits and read the action bits as an action (decoded to the nearest code, or directly for a register). One circuit per distinct validation state code is run with 1024, 16384 or 32768 shots, mitigated by dynamical decoupling (Viola et al., 1999) and twirling (Wallman & Emerson, 2016), and $q(a \mid s)$ is the histogram of the decoded actions with a pseudo-count of $1/82$ per action. We fit a per-qubit asymmetric bit-flip channel of the quantum processor by regressing the measured action-bit marginals on the exact ones. It drives the noise-aware string assignment and predicts a register model's quantum-processor policy by pushing its exact string distribution through the channel.

Table 1: The model from the closed form to the quantum processor (816 validation rows). CE is the cross-entropy (3), top-k the fraction of rows whose teacher arg-max is among the model's k most probable actions. S/A/H counts state, action and hidden units, and "reg." marks a fully assigned register. QITE limit denotes the exact QITE step on whole components (Section 4.3), 5-layer QAOA the tuned five-layer circuit, and DD+Twirl dynamical decoupling and twirling. Runs use 1024 shots per state unless marked "16k" or "32k". A "fixed" state code keeps its random initialisation, a state code "learned π^2" is learned on sharpened targets, "(full)" with the full code-learning schedule, and "refitted" the model retrained with strings for the channel refitted on the 32k QITE-limit run. Scenario-level bootstrap standard errors are about 0.05 nats for CE and 0.04 for top-k per row. Rows below the second rule are QPU runs, and their last column gives the exact top-1 of the model. Bold marks the best QPU value and underlining the best classical value.

model	preparation / read-out	action code	state code	S/A/H	CE	top-1	top-2	top-3	exact
baseline	state-independent mean policy				3.781	0.270	0.479	0.569	
dense	closed form, exact	index 7	one-bit	20/7/93	3.748	0.398	0.502	0.565	
dense	closed form + L-BFGS, exact	index 7	one-bit	20/7/93	3.398	0.577	0.733	0.816	
dense	learned code, exact	two-hot 14	learned	20/14/86	<u>3.193</u>	<u>0.703</u>	<u>0.850</u>	<u>0.906</u>	
fan-in-4	learned code, exact	two-hot 14	learned	20/14/86	3.502	0.560	0.699	0.783	
grid RBM	learned code, exact	random 36	learned	22/36/60	3.429	0.627	0.746	0.806	
grid RBM	classical Gibbs, nearest code	random 36	learned	22/36/60	4.080	0.391	0.450	0.507	
grid RBM	QAOA-based layer	reg. 8	learned	50/8/60	3.954	0.270	0.479	0.537	0.353
grid RBM	edge-local QITE 5×0.1, DD+Twirl	reg. 8	learned	50/8/60	3.837	0.270	0.482	0.544	0.353
grid RBM	QITE limit, DD+Twirl	reg. 8	learned	50/8/60	3.686	0.349	0.531	0.637	0.353
grid RBM	QITE limit, DD+Twirl, 16k	reg. 10, noise-aware	fixed	48/10/60	3.689	0.419	0.479	0.571	0.424
grid RBM	QITE limit, DD+Twirl, 16k	reg. 10, noise-aware	learned	48/10/60	3.619	0.458	0.515	0.631	0.458
grid RBM	QITE limit, DD+Twirl, 16k	reg. 10, noise-aware	learned π^2	48/10/60	3.649	0.484	0.637	0.686	0.488
grid RBM	QITE limit, DD+Twirl, 16k	reg. 12, noise-aware	learned π^2	46/12/60	3.566	0.488	0.662	0.734	0.494
grid RBM	QITE limit, DD+Twirl, 32k	reg. 12, noise-aware	learned π^2 (full)	46/12/60	**3.551**	**0.494**	0.680	0.748	0.512
grid RBM	5-layer QAOA, DD+Twirl, 16k	reg. 12, noise-aware	learned π^2 (full)	46/12/60	3.630	0.369	0.582	0.659	0.512
grid RBM	QITE limit, DD+Twirl, 32k	reg. 12, refitted	learned π^2 (full)	46/12/60	3.555	0.484	**0.691**	**0.768**	0.490

5 EXPERIMENTS

Table 1 traces the model from the dense closed form to the best policy read out of the quantum processor. Consecutive rows differ in the ingredients named in their columns, mostly one at a time. Standard errors are about 0.05 nats for CE and 0.04 for top-k per row (Appendix E), so single steps of a few points are trends that the ladder as a whole confirms. Classical models are read out exactly by the softmax over their action codes, and the QPU rows are runs on `ibm_phoenix`. All runs of this paper took 63 minutes of quantum processor time.

5.1 CLASSICAL MODELS

Three trends carry through the classical part of the table. First, the state code learned jointly with the model is worth more than any other ingredient: it brings the dense RBM within a fifth of a nat of the teacher's entropy and above 0.7 top-1. Second, adapting the codes to the structure limits its cost. The pruned fan-in-4 model keeps an eighth of the weights and loses fourteen top-1 points. Third, a structure fixed in advance beats pruning to the same fan-in. The grid RBM is better than the pruned fan-in-4 model on every metric, because its codes were designed for the structure instead of surviving it. Training and validation cross-entropy differ by a few hundredths in either direction, so the gaps are capacity, not overfitting. For scale, logistic regression on the 20 loadings reaches 3.29 / 0.62 and a 3000-parameter perceptron 3.13 / 0.79 (Appendix E): the RBM is not the strongest student but the one whose Gibbs state a quantum processor can hold. The classical sampling row shows the problem that the register solves. Exact Gibbs samples of the grid RBM, decoded to the nearest action code, lose a third of the agreement of the exact read-out, because a sample of the 36 action bits is almost never an admissible code. All runs of the 36-bit model on the quantum processor stay below a top-1 of 0.1 (all runs in Appendix E). The QPU rows of Table 1 therefore use action registers.

5.2 Runs on the Quantum Processor

The single-layer circuit returns the state-independent prior for both registers, as classical samples at its effective temperature predict (Appendix E). Tuned angles do not change this, and five tuned layers reach a top-1 of 0.37 at five times the gates of the QITE-limit step. The edge-local evolution stays at the prior as well, because a two-qubit domain cannot reweight the other leaves of a star. One QITE-limit step per star changes the picture, and the 8-bit register beats the prior.

Once the strings protect the currently worst qubits, the remaining gap between a quantum-processor policy and its exact read-out is shot noise: the fitted channel costs a few thousandths of a nat and 16384 mitigated shots reproduce every exact read-out to a few hundredths. Noise-aware strings add four top-1 points over plain ones at the same fixed state code, a code effect since the channel costs both the same. Learning the state code for the register adds another four points. Sharpening the target to π^2 adds two points of top-1 and seven of top-2 at a fixed state code and unchanged cross-entropy, and learning the code on π^2 adds about one and five more on the QPU for 0.03 nats. Wider registers leave top-1 on a plateau: the 14-bit register with its own learned code matches the 12-bit one in cross-entropy and top-3 and loses two points of exact top-1, so width improves calibration but not the arg-max. Learning the state code for the 12-bit register on π^2 lifts every metric, and with the full schedule its exact read-out is the first register model above half agreement with the teacher. On the quantum processor the 12-bit models are the best quantum-processor policies of this work on every metric, agreeing with the teacher in about half of the held-out states. The full-schedule model returns 0.483 to 0.494 in three runs, so its two points below the exact read-out are systematic. The flip channel refitted on this run (Appendix E) traces them to an action qubit whose flip rate had risen to $31\,\%$. Retraining with strings for the refitted channel removes the loss but costs exact capacity: the retrained model reads out within shot noise of its exact read-out, whose top-1 of 0.490 now caps the agreement two points lower, with the best top-2 and top-3 of the QPU runs. The quantum processor therefore holds and reads out the conditional Gibbs state of a 118-qubit RBM at the model's own temperature. That state is a product of five-qubit stars with classically compiled angles, so the experiment tests preparation and read-out, not sampling hardness. The samples are joint: over 22 231 action–hidden pairs the measured connected correlations match the exact ones with correlation 0.98, where single-qubit rotations would give zero (Appendix E). The QPU is currently not the bottleneck. What separates the agreement from the 0.63 of the exact 36-bit grid RBM is the capacity lost to the register, a property of the classical model. This leaves much room for future research on quantum hardware-aware loss functions and model classes: objectives that account for the register and the measured noise during training, and models whose read-out is exact by construction. Sampled without clamping, with two-qubit gates on all 210 couplers and all 118 qubits measured, the single layer gives a state whose magnetisations and connected correlations follow the grid RBM tempered to $\beta = 0.18$ with correlation 0.91 and 0.96 (Appendix E).

6 Conclusion

We have distilled a neural network control policy, the teacher, into an RBM student whose model space is constrained by the quantum processor: its graph is the lattice of every functioning qubit and coupler of a 120-qubit Nighthawk r2 QPU, its action code follows the measured qubit noise, and its state code is learned for that register. Sampled on the quantum processor, its read-out recovers the trained model. First, the learned state encoding was worth more than any other ingredient, and training on the lattice beat pruning to it. Second, the action code decides what can be read out. The dense random code is unreadable from samples, and a register is readable but pays in capacity, which noise-aware strings, sharpened targets and a register-specific state code recover. Third, the outcome was decided by the preparation of the Gibbs state rather than by the noise of the quantum processor. QAOA-type circuits and edge-local imaginary-time steps stay at or near the prior, whereas the QITE limit is exact at under a hundred two-qubit gates and, with strings matched to the current channel, loses only to shot noise.

We chose the model so that its results can be validated. The experiments show that the quantum processor holds a 118-qubit RBM at its own temperature and reads out joint samples of its action and hidden units, and they measure what each step towards a QPU is worth. The same steps apply to models whose inference is classically intractable and whose action bits stay correlated given the state.

Reproducibility statement

Sections 2–4 define the data split, the metrics, every model and every circuit. Appendices A–E list the model-selection studies and all sampling runs with their settings, and Appendix H the implementation. The supplement contains the trained grid models, the decision log, the Ising forms, all circuit and analysis scripts and the records of every run on the quantum processor, with instructions for rerunning the circuits. The training code depends on a library that will be released with the camera-ready version.

AI use statement

Large language models were used as a coding and writing assistant, for drafting and editing parts of the text and of the analysis scripts. All of it was checked and tested by the authors. The research questions, methods and experiments are the authors' own.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

Steven H. Adachi and Maxwell P. Henderson. Application of quantum annealing to training of deep neural networks. *arXiv:1510.06356*, 2015.

Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum Boltzmann machine. *Physical Review X*, 8:021050, 2018.

Eric R. Anschuetz and Yudong Cao. Realizing quantum Boltzmann machines through eigenstate thermalization. *arXiv:1903.01359*, 2019.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz. Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning. *Physical Review A*, 94:022308, 2016.

Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz. Quantum-assisted learning of hardware-embedded probabilistic graphical models. *Physical Review X*, 7: 041052, 2017.

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

Kelly Boothby, Paul Bunyk, Jack Raymond, and Aidan Roy. Next-generation topology of D-Wave quantum processors. *arXiv:2003.00133*, 2020.

Kelly Boothby, Andrew D. King, and Jack Raymond. Zephyr topology of D-Wave quantum processors. Technical Report 14-1056A-A, D-Wave Systems, 2021.

Fernando G. S. L. Brandão and Michael J. Kastoryano. Finite correlation length implies efficient preparation of quantum thermal states. *Communications in Mathematical Physics*, 365:1–16, 2019.

Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 535–541, 2006.

P. I. Bunyk, Emile M. Hoskinson, Mark W. Johnson, Elena Tolkacheva, Fabio Altomare, Andrew J. Berkley, Richard Harris, Jeremy P. Hilton, Trevor Lanting, Anthony J. Przybysz, and Jed Whittaker. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions on Applied Superconductivity*, 24(4):1700110, 2014.

João Caldeira, Joshua Job, Steven H. Adachi, Brian Nord, and Gabriel N. Perdue. Restricted Boltzmann machines for galaxy morphology classification with a quantum annealer. *arXiv:1911.06259*, 2020.

Christopher Chamberland, Guanyu Zhu, Theodore J. Yoder, Jared B. Hertzberg, and Andrew W. Cross. Topological and subsystem codes on low-degree graphs with flag qubits. *Physical Review X*, 10:011022, 2020.

Chi-Fang Chen, Michael J. Kastoryano, and András Gilyén. An efficient and exact noncommutative quantum Gibbs sampler. *arXiv:2311.09207*, 2023.

Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of Trotter error with commutator scaling. *Physical Review X*, 11:011020, 2021.

Vicky Choi. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Information Processing*, 7:193–209, 2008.

Anirban N. Chowdhury and Rolando D. Somma. Quantum algorithms for Gibbs sampling and hitting-time estimation. *Quantum Information and Computation*, 17(1–2):41–64, 2017.

Luuk Coopmans and Marcello Benedetti. On the sample complexity of quantum Boltzmann machine learning. *Communications Physics*, 7:274, 2024.

Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Simmons, and Seyon Sivarajah. On the qubit routing problem. In *14th Conf. Theory of Quantum Computation, Communication and Cryptography (TQC), LIPIcs 135*, pp. 5:1–5:32, 2019.

D-Wave Systems. Errors and error correction: Integrated control errors. D-Wave System Documentation, `https://docs.dwavesys.com/docs/latest/c_qpu_ice.html`, 2024.

D-Wave Systems. Minor-embedding: Best practices. D-Wave System Documentation, `https://docs.dwavequantum.com/en/latest/quantum_research/embedding_guidance.html`, 2025.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

Pablo Díez-Valle, Diego Porras, and Juan José García-Ripoll. Quantum approximate optimization algorithm pseudo-Boltzmann states. *Physical Review Letters*, 130:050601, 2023. arXiv:2201.03358.

Vivek Dixit, Raja Selvarajan, Muhammad A. Alam, Travis S. Humble, and Sabre Kais. Training restricted Boltzmann machines with a D-Wave quantum annealer. *Frontiers in Physics*, 9:589626, 2021.

Vivek Dixit, Raja Selvarajan, Tamer Aldwairi, Yaroslav Koshka, Mark A. Novotny, Travis S. Humble, Muhammad A. Alam, and Sabre Kais. Training a quantum annealing based restricted Boltzmann machine on cybersecurity data. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):417–428, 2022.

Vincent Dumoulin, Ian J. Goodfellow, Aaron Courville, and Yoshua Bengio. On the challenges of physical implementations of RBMs. In *Proc. 28th AAAI Conf. Artificial Intelligence*, pp. 1199–1205, 2014.

Abdelmoula El-Yazizi and Yaroslav Koshka. Investigation of D-Wave quantum annealing for training restricted Boltzmann machines and mitigating catastrophic forgetting. *arXiv:2508.15697*, 2025.

Abdelmoula El Yazizi, Samee U. Khan, and Yaroslav Koshka. Comparison of D-Wave quantum annealing and Markov chain Monte Carlo for sampling from a probability distribution of a restricted Boltzmann machine. *arXiv:2508.10228*, 2025.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. In *arXiv preprint arXiv:1411.4028*, 2014.

Asja Fischer and Christian Igel. Training restricted Boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39, 2014.

Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv:quant-ph/0208112*, 2002.

W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Jared B. Hertzberg, Eric J. Zhang, Sami Rosenblatt, Easwar Magesan, John A. Smolin, Jeng-Bang Yau, Vivekananda P. Adiga, Martin Sandberg, Markus Brink, Jerry M. Chow, and Jason S. Orcutt. Laser-annealing Josephson junctions for yielding scaled-up superconducting quantum processors. *npj Quantum Information*, 7:129, 2021.

Catherine F. Higham and Adrian Bedford. Quantum deep learning by sampling neural nets with a quantum annealer. *Scientific Reports*, 13:3939, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in Neural Information Processing Systems 29*, pp. 4107–4115, 2016.

IBM Quantum. IBM Quantum Nighthawk processors. `https://quantum.cloud.ibm.com`, 2026.

Seon-Geun Jeong, Mai Dinh Cong, Dae-Il Noh, Quoc-Viet Pham, and Won-Joo Hwang. Embedding-aware noise modeling of quantum annealing. *arXiv:2510.04594*, 2025.

Joshua Job and Steve Adachi. Systematic comparison of deep belief network training using quantum annealing vs. classical techniques. *arXiv:2009.00134*, 2020.

Mārtiņš Kālis, Andris Locāns, Rolands Šikovs, Hassan Naseri, and Andris Ambainis. A hybrid quantum-classical approach for inference on restricted Boltzmann machines. *Quantum Machine Intelligence*, 5:44, 2023.

Mária Kieferová and Nathan Wiebe. Tomography and generative training with quantum Boltzmann machines. *Physical Review A*, 96:062327, 2017.

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

Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P. Orlando, Simon Gustavsson, and William D. Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6: 021318, 2019.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory*, 47(2):498–519, 2001.

Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted Boltzmann machines. In *Proc. 25th Int. Conf. Machine Learning (ICML)*, pp. 536–543, 2008.

Hugo Larochelle, Michael Mandel, Razvan Pascanu, and Yoshua Bengio. Learning algorithms for the classification restricted Boltzmann machine. *Journal of Machine Learning Research*, 13:643–669, 2012.

David Layden, Guglielmo Mazzola, Ryan V. Mishmash, Mario Motta, Pawel Wocjan, Jin-Sung Kim, and Sarah Sheldon. Quantum-enhanced Markov chain Monte Carlo. *Nature*, 619:282–287, 2023.

Gushu Li, Yufei Ding, and Yuan Xie. Tackling the qubit mapping problem for NISQ-era quantum devices. In *Proc. 24th Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 1001–1014, 2019.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. Quantum inference on Bayesian networks. *Physical Review A*, 89:062315, 2014.

Antoine Marot, Benjamin Donnot, Camilo Romero, Balthazar Donon, Marvin Lerousseau, Luca Veyrin-Forrer, and Isabelle Guyon. Learning to run a power network challenge for training topology controllers. In *Electric Power Systems Research*, volume 189, 2020.

Antoine Marot, Benjamin Donnot, Gabriel Dulac-Arnold, Adrian Kelly, Aidan O'Sullivan, Jan Viebahn, Mariette Awad, Isabelle Guyon, Patrick Panciatici, and Camilo Romero. Learning to run a power network challenge: a retrospective analysis. In *Proc. NeurIPS 2020 Competition and Demonstration Track, PMLR 133*, pp. 112–132, 2021.

Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, 5:75, 2019.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Lorenzo Moro and Enrico Prati. Anomaly detection speed-up by quantum restricted Boltzmann machines. *Communications Physics*, 6:269, 2023.

Mario Motta, Chong Sun, Adrian T. K. Tan, Matthew J. O'Rourke, Erika Ye, Austin J. Minnich, Fernando G. S. L. Brandão, and Garnet Kin-Lic Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics*, 16:205–210, 2020.

Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. Transformation of quantum states using uniformly controlled rotations. *Quantum Information and Computation*, 5 (6):467–473, 2005.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14:265–294, 1978.

Hirofumi Nishi, Taichi Kosugi, and Yu-ichiro Matsushita. Implementation of quantum imaginary-time evolution method on NISQ devices by introducing nonlocal approximation. *npj Quantum Information*, 7:85, 2021.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1 (3):127–239, 2014.

Emanuel Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Thomas Pochart, Paulin Jacquot, and Joseph Mikael. On the challenges of using D-Wave computers to sample Boltzmann random variables. *arXiv:2111.15295*, 2021.

John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

Patrick Rall, Chunhao Wang, and Pawel Wocjan. Thermal state preparation via rounding promises. *Quantum*, 7:1132, 2023.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Çaglar Gülçehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. In *Int. Conf. Learning Representations (ICLR)*, 2016.

Ramiro Sagastizabal, Shavindra P. Premaratne, Bart A. Klaver, Martin A. Rol, Viorel Negîrneac, Matheus S. Moreira, et al. Variational preparation of finite-temperature states on a quantum computer. *npj Quantum Information*, 7:130, 2021.

Hadi Salloum, Laith Nayal, and Manuel Mazzara. Evaluating the Advantage 2 quantum annealer prototype: A comparative evaluation with Advantage 1 and hybrid solver and classical restricted Boltzmann machines on MNIST classification. In *Mathematical Modeling in Physical Sciences (Springer Proc. Math. Stat.)*, pp. 371–384, 2025.

Ilmo Salmenperä and Jukka K. Nurminen. Software techniques for training restricted Boltzmann machines on size-constrained quantum annealing hardware. *Frontiers in Computer Science*, 5: 1286591, 2023.

David Silver, Julian Schrittwieser, Karen Simonyan, et al. Mastering the game of Go without human knowledge. *Nature*, 550:354–359, 2017.

Salvatore Sinno, Markus Bertl, Arati Sahoo, Bhavika Bhalgamiya, Thomas Groß, and Nicholas Chancellor. Implementing large quantum Boltzmann machines as generative AI models for dataset balancing. In *Proc. Int. Conf. Next Generation Information System Engineering (NGISE)*, pp. 1–8, 2025.

Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In David E. Rumelhart and James L. McClelland (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, pp. 194–281. MIT Press, 1986.

Michael Streif and Martin Leib. Training the quantum approximate optimization algorithm without access to a quantum processing unit. *Quantum Science and Technology*, 5:034008, 2020.

Masuo Suzuki. Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics*, 51:183–190, 1976.

Graham W. Taylor, Geoffrey E. Hinton, and Sam T. Roweis. Modeling human motion using binary latent variables. In *Advances in Neural Information Processing Systems 19*, pp. 1345–1352, 2007.

Kristan Temme, Tobias J. Osborne, Karl G. Vollbrecht, David Poulin, and Frank Verstraete. Quantum Metropolis sampling. *Nature*, 471:87–90, 2011.

Guillaume Verdon, Michael Broughton, and Jacob Biamonte. A quantum algorithm to train neural networks using low-depth circuits. *arXiv:1712.05304*, 2017.

Lorenza Viola, Emanuel Knill, and Seth Lloyd. Dynamical decoupling of open quantum systems. *Physical Review Letters*, 82:2417–2421, 1999.

Joel J. Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94:052325, 2016.

Jingxiang Wu and Timothy H. Hsieh. Variational thermal quantum simulation via thermofield double states. *Physical Review Letters*, 123:220502, 2019.

Yusen Wu, Chunyan Wei, Sujuan Qin, Qiaoyan Wen, and Fei Gao. Quantum restricted Boltzmann machine universal for quantum computation. *arXiv:2005.11970*, 2020.

Daiwei Zhu, Sonika Johri, Norbert M. Linke, Kevin A. Landsman, C. Huerta Alderete, Nhung H. Nguyen, et al. Generation of thermofield double states and critical ground states with a quantum computer. *Proceedings of the National Academy of Sciences*, 117:25402–25406, 2020.

Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Variational quantum Boltzmann machines. *Quantum Machine Intelligence*, 3:7, 2021.

Alwin Zulehner, Alexandru Paler, and Robert Wille. An efficient methodology for mapping quantum circuits to the IBM QX architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(7):1226–1236, 2019.

A CLOSED-FORM VARIANTS AND PROTOTYPE SELECTION

This appendix collects the closed-form results that Sections 3.2 and 3.3 refer to: hand-made state codes and action codes under the pure closed form (Table 2), the comparison of prototype selection rules (Table 3) and the check of the greedy rule against simulated annealing (Table 4). All numbers are for the split with scales chosen by cross-validation ($\alpha = 3$, $C = 6$ unless noted). No fine-tuning is involved.

A.1 THE CLOSED FORM AS A MAXIMUM-LIKELIHOOD LIMIT

The construction of Section 3.2 is non-standard, so we record what it estimates. Let the visible layer be $v \in \{0,1\}^n$ (state and action bits together), let the training set have M distinct configurations $\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(M)}$ with empirical masses p_h ($\sum_h p_h = 1$), and give the RBM one hidden unit per configuration with the weights (5) and full connectivity, $N(h) = \{1, \dots, n\}$, $\eta = \xi = 0$. The activation of unit h at a visible configuration v is then $a_h(v) = b_h + \sum_u W_{uh} v_u = C + \log p_h - \alpha\, d_{\mathrm{H}}(v, \mathbf{z}^{(h)})$, the matched-filter identity, and summing out the hidden units gives the visible marginal

$$p(v) = \frac{1}{Z} \prod_{h=1}^{M} \Big(1 + e^{C}\, p_h\, e^{-\alpha\, d_{\mathrm{H}}(v, \mathbf{z}^{(h)})}\Big). \tag{8}$$

For $\alpha \to \infty$ every factor with $d_{\mathrm{H}}(v, \mathbf{z}^{(h)}) > 0$ tends to 1, so $p(v) = (1 + e^{C} p_{h(v)})/Z$ if $v = \mathbf{z}^{(h(v))}$ is a training configuration and $p(v) = 1/Z$ otherwise, with $Z = 2^n + e^{C}$. Hence

$$p(v) = (1 - \lambda)\, \hat{p}(v) + \lambda\, 2^{-n}, \qquad \lambda = \frac{2^n}{2^n + e^{C}}, \tag{9}$$

where $\hat{p}(v) = p_{h(v)}$ is the empirical distribution of the training set: the closed form is the empirical distribution mixed with a uniform background of weight λ, and $\lambda \to 0$ as $C \to \infty$. Since the empirical distribution maximises the likelihood $\sum_i \log p(v_i)$ over all distributions on $\{0,1\}^n$ (Gibbs' inequality), the construction with one hidden unit per distinct configuration attains the maximum-likelihood estimate in the limit $\alpha, C \to \infty$. At finite C it is the additively smoothed estimate, and at finite α every configuration is in addition spread over its Hamming neighbourhood by the exponential kernel, whose first-order expansion of (8) in $e^{C} p_h e^{-\alpha d_{\mathrm{H}}}$ is the Parzen estimate quoted in Section 3.2. The conditional read-out inherits the statement: $q(a \mid s) = p(\mathbf{x}(s), c(a)) / \sum_{a'} p(\mathbf{x}(s), c(a'))$ tends to the empirical conditional $\hat{\pi}(a \mid s)$, the minimiser of the conditional cross-entropy (3) over all conditional distributions, i.e. the maximum-likelihood estimate of the conditional. With fewer hidden units than distinct configurations the prototype selection and the mass merge of Section 3.2 replace $\hat{p}$ by a quantised empirical distribution, and fine-tuning moves the weights away from this limit towards the regularised optimum of (3). The closed form is therefore a consistent starting point rather than the final estimate.

A.2 CLOSED FORM AND PROTOTYPE SELECTION

This subsection compares the closed-form variants and the prototype selection rules that led to the closed-form rows of Table 1.

Closed-form variants Table 2 shows the pure closed form with hand-made codes and different action codes. The unit budget costs little: 93 hidden units score 3.614 against 3.593 for a diagnostic model with one unit per configuration. The limiting factor is the information in the binarised loadings and the intrinsic randomness of the policy.

Selection rules compared We compared *topmass* (the M heaviest configurations), *facility* (greedy kernel coverage of the mass), kernel *herding* (greedy maximum-mean-discrepancy (MMD) matching), k*-centre* (farthest point) and the task-aware *cegreedy* rule, forward selection of the configuration whose closed-form unit most reduces the training conditional cross-entropy (3). After selection, every configuration's mass is merged into its nearest prototype (in the α-weighted Hamming distance), so the p_h sum to one. Table 3 shows that at a sharp kernel the mass-based rules pick almost the same prototypes, diversity-first rules pick low-mass configurations and hurt, and only cegreedy differs materially: it fixes $q(a \mid s)$ where the policy is peaked and buys three to four points of top-1 and top-3 agreement at equal cross-entropy.

Table 2: Closed-form models without fine-tuning (one-bit state code $\rho_j \geq 0.8$ unless noted, scales by cross-validation).

variant	vis.	hid.	CE	top-1
one-bit, 7-bit index, cegreedy	27	93	**3.614**	**0.542**
one-bit, index, facility	27	93	3.619	0.506
one-bit, 12-bit spread action code	32	88	3.618	0.480
one-bit at 0.9, soft merge	27	93	3.641	0.462
two thresholds per line	47	73	3.802	0.538
one-bit, one-hot actions	102	18	3.833	0.270
diagnostic: all 6000 configurations	27	6000	3.593	0.558
baseline: mean policy			3.781	0.270

Table 3: Prototype selection rules (one-bit code, index actions, hard merge, i.e. each configuration's mass moved to its nearest prototype, $\alpha = 3$, $C = 6$).

rule	principle	CE	top-1	top-2	top-3
cegreedy	CE forward selection	**3.614**	**0.542**	0.625	**0.737**
facility	kernel coverage	3.619	0.506	**0.629**	0.702
topmass	heaviest configurations	3.619	0.506	0.625	0.684
herding	greedy MMD	3.639	0.502	0.589	0.664
k-centre	farthest point ($\alpha = 2$)	4.098	0.235	0.385	0.390
baseline	mean policy	3.781	0.270	0.479	0.569

Table 4: Greedy versus simulated annealing (SA) for prototype selection ($\alpha = 3$, $C = 6$). "own": prototypes' own masses. "merged": masses of the final model.

objective	start	moves	objective start $\rightarrow$ SA	val CE
kernel MMD	herding	1M	$0.002424 \rightarrow 0.002426$	3.639
kernel MMD	random	1M	$-0.0084 \rightarrow 0.002426$	3.638
CE (own)	cegreedy	100k	$3.6910 \rightarrow 3.6910$	3.614
CE (own)	random	300k	$4.35 \rightarrow 3.6910$	3.614–3.623
CE (own)	topmass	100k	$3.6966 \rightarrow 3.6911$	3.614
CE (merged)	cegreedy	20k	$3.6473 \rightarrow 3.6473$	3.614
CE (merged)	random	20k	$4.42 \rightarrow 3.6613$	3.621

Is greedy good enough? Selection is a combinatorial problem over $\mathbf{z} \in \{0,1\}^K$, $\sum z_i = M$. We optimised two objectives by simulated annealing with swap moves (temperature calibrated from sampled swap deltas, geometric schedule): the quadratic kernel-MMD objective, whose greedy is herding, and the exact training cross-entropy of the RBM conditional (Table 4). Annealing never improved on the greedy set: from random starts it came within 0.015 nats of the greedy solution for both objectives, and started at the greedy set it found no improving neighbourhood. The greedy is therefore at or extremely close to the optimum of these objectives, and the remaining gap to the policy is a property of the objective and the code, not of the optimiser. The quadratic MMD surrogate matches the density rather than the conditional and is the poorer proxy.

B Fine-Tuning and the Learned Code

Table 5 shows the effect of fine-tuning and of the learned code, Table 6 the design study of the learned binarisation. The default pipeline (random start, 20 bits, 15 rounds of 600 iterations, 1500 prototype candidates, L-BFGS 3000) reaches $\mathrm{CE} = 3.247$ and top-1 0.724 on the split and 3.314 / 0.740 / 0.884 / 0.921 (top-1/2/3) on a second, 887-row split, without any adaptive-step method. Controlled comparisons behind the optimiser choice: L-BFGS 1000 iterations equal FISTA 4000 (CE 3.230 vs 3.224) in a third of the time. For the encoder, backtracking FISTA (3.220) beat gradient descent (3.224) and fixed-step FISTA (3.231 at step 0.5, 3.263 at step 2, which diverged on raw features).

Table 5: Closed form, fine-tuning (L-BFGS, 3000 iterations) and learned codes (7-bit index actions unless noted, $\alpha = 1.5$, $C = 6$, cegreedy).

state code and fit	CE	top-1	top-2	top-3
one-bit at 0.8, closed form	3.748	0.398	0.502	0.565
one-bit at 0.8, + L-BFGS	3.398	0.577	0.733	0.816
learned, one-bit start, + L-BFGS	3.204	0.694	0.855	0.901
learned, random start (default), + L-BFGS	3.247	**0.724**	**0.863**	0.890
learned, one-bit start, two-hot, + L-BFGS	**3.193**	0.703	0.850	**0.906**
baseline: mean policy	3.781	0.270	0.479	0.569

Table 6: Design study of the learned binarisation (816 rows, 15 rounds = 1 hard + 12 soft (κ: $4 \to 30$) + 2 hard, 600 iterations each, L-BFGS 3000 final, unless noted). "states": distinct training states of the code.

init	bits	schedule	cand.	CE	top-1	top-3	states
one-bit	20	9×300	1500	3.217	0.662	0.896	261
one-bit	20	15×600	1500	3.205	0.706	0.909	327
one-bit	20	15×600	all	**3.204**	0.694	0.901	327
one-bit	20	21×800	all	3.278	0.643	0.882	297
$\mathcal{N}(0,1)$	16	15×600	1500	3.247	0.689	0.881	546
$\mathcal{N}(0,1)$	20	15×600	1500	3.250	0.701	0.895	687
$\mathcal{N}(0,1)$	24	15×600	1500	3.225	0.701	0.909	864
$\mathcal{N}(0,1)$	28	15×600	1500	3.253	0.719	0.892	1059
$\mathcal{N}(0,2)$ (default)	20	15×600	1500	3.247	0.724	0.890	679
$\mathcal{N}(0,2)$, seed 2	20	15×600	all	3.280	0.662	0.865	757
$\mathcal{N}(0,2)$	24	15×600	all	3.292	0.686	0.859	853
$\mathcal{N}(0,3)$	24	15×600	all	3.264	**0.728**	0.882	904
default pipeline, fresh seed (887-row split)				3.314	0.740	0.921	704
one-bit code, closed form + L-BFGS (same split)				3.467	0.604	0.827	249

Table 7: Fan-in-limited RBMs with two-hot actions (20+14+86 units), dense FISTA pre-training, 14 continuation rounds (λ: $10^{-4} \to 0.1$), projection, re-coding rounds against the projected model, masked L-BFGS.

state code	action code	K	edges	CE	top-1	top-3
learned (one-bit start)	two-hot	8	688	3.365	0.582	0.830
+ re-coded	two-hot	8	688	**3.337**	**0.626**	0.832
learned (random start) + re-coded	two-hot	8	688	3.384	0.582	0.767
learned (one-bit start)	two-hot	4	344	3.560	0.499	0.695
+ re-coded	two-hot	4	344	3.501	0.526	0.749
learned (one-bit, 15×600) + re-coded	two-hot	4	344	**3.502**	**0.560**	**0.783**
same, at least $K_a = 1$ action edge per unit	two-hot	4	344	3.502	0.551	0.724
learned (random start) + re-coded	two-hot	4	344	3.510	0.505	0.643
learned, two-hot, dense reference	two-hot	–	2924	3.193	0.703	0.906

C FAN-IN-LIMITED MODELS

Table 7 collects the sparse models. With $K = 4$ the best model keeps 344 of 2924 weights (12 %) and loses about 0.3 nats and 14 top-1 points against the dense two-hot model, but stays clearly above the one-bit closed form (3.75 / 0.40) and the baseline. $K = 8$ (25 % of the weights) closes about half of the gap to the dense model. The per-group minimum $K_a = 1$ removes the seven dead units and improves the pre-recoding cross-entropy (3.651 vs 3.682) but the final models are indistinguishable within noise.

Table 8: Grid-embedded RBMs on the 12×10 grid without the defect vertices 17 and 62: 118 units, 210 edges, fan-in 2–4 (816 validation rows). The eight most promising configurations of a wider search on a grid with a single defect. "init code": closed form + L-BFGS on the initial state code before code learning. "learned": after 15 code-learning rounds and L-BFGS 3000. S/A/H = state, action and hidden units. The 44-bit model leaves 9 hidden units without a state neighbour.

action code	vis. class	S/A/H	state init	C	init code CE / top-1	learned CE	top-1	top-2	top-3
random 36	1	24/36/58	random	4	3.551 / 0.466	**3.425**	0.596	**0.752**	**0.838**
random 44	0	14/44/60	random	4	3.548 / 0.500	3.429	0.587	0.729	0.781
random 36	0	22/36/60	random	4	3.545 / 0.451	3.429	**0.627**	0.746	0.806
random 28	0	30/28/60	random	4	3.588 / 0.462	3.446	0.591	0.730	0.803
twohot 36	0	22/36/60	random	4	3.662 / 0.373	3.453	0.548	0.667	0.795
twohot 28	0	30/28/60	random	4	3.720 / 0.407	3.513	0.562	0.722	0.786
twohot 20	1	40/20/58	random	6	3.908 / 0.442	3.691	0.618	0.678	0.705
twohot 20	1	40/20/58	random	4	3.904 / 0.436	3.699	0.591	0.675	0.708

D GRID SEARCH

Table 8 shows the eight configurations evaluated on the grid without vertices 17 and 62 (all 118 units and 210 edges, and in every model but the 44-bit one every hidden unit has at least one state and one action neighbour). They are the eight best of a wider search of fourteen configurations (closed-form constant $C \in \{2, 4, 6\}$ of (5), threshold and random state-code starts, both colour classes, 20 to 44 action bits, two-hot and dense random codes) run on the grid with vertex 17 as the only defect. That search showed that the boost, the state-code start and the colour class change the results by at most 0.04 nats, whereas the action code and its length dominate. Three results carry over.

The action code decides. With the two-hot code the grid model stays at CE ≥ 3.45 and top-1 ≤ 0.62 whatever the number of action bits. A dense random code with half of the action bits on per action improves every configuration with the same number of action bits by 0.02 to 0.07 nats ($3.453 \to 3.429$ at 36 bits, $3.513 \to 3.446$ at 28 bits) and by up to 8 top-1 points. A hidden unit with one or two action neighbours then splits the action set in half per bit instead of marking 8 of 82 actions, and the fine-tuning can exploit that even through the local closed form. *More action bits than state bits.* Moving visible vertices from the state code to the action code helps up to 36 action bits (22 learned state bits). At 44 bits the 14 remaining state bits no longer cover all hidden units and top-2 and top-3 drop. *The colour classes are equivalent within noise*: with 36 random bits, 60 hidden units give 3.429 / 0.627 and 58 hidden units 3.425 / 0.596.

The selected model (random 36-bit action code, 22 state bits, 60 hidden units, $C = 4$, see Figs. 1 and 2) reaches CE $= 3.429$ and top-1/2/3 $= 0.627/0.746/0.806$. Its hidden units see 1, 2 or 3 action bits (6, 41 and 13 units) and 1 or 2 state bits (37 and 23 units). This is better on every metric than the pruned fan-in-4 RBM of Section 3.4 (3.502 / 0.560 / 0.699 / 0.783), which has up to the same number of connections per hidden unit (four, against two to four) but 134 more edges (344): a structure fixed in advance, with the code and the placement designed around it, beats a dense model pruned down to the same fan-in. The training cross-entropy (3.47 after the last code-learning round) stays within a few hundredths of the validation value in all grid runs, so the remaining gap to the dense 120-unit RBM (3.247 / 0.724 for the random-start model of this study, 3.193 / 0.703 for the two-hot dense model of Table 1) is the capacity of the planar structure, not overfitting.

E SAMPLING EXPERIMENTS OUTSIDE TABLE 1

Table 9 lists every register run on the quantum processor, Table 11 the runs of the 36-bit grid model, all below top-1 0.1, and Table 10 the exact read-outs of the register models (their grid embeddings are shown in Figs. 3, 4 and 5), the noiseless emulations of their circuits and the sample-based read-outs of the grid models obtained with classical samplers on the same conditional Ising model as the quantum-processor runs of Section 5.2: the state qubits are clamped, the hidden and action units are sampled, the hidden units are discarded and the action bits are read out exactly as for the quantum-processor samples. Three families of models recur below: the 36-bit grid RBM with the dense random action code, the strongest exact grid model of Table 1, the 8- and 10-bit register models, the

Table 9: All register runs on the quantum processor (816 validation rows), read out as in Table 1: one circuit per distinct validation state code, hidden qubits discarded, every measured register string an action. Preparation: QAOA-based layer (angles by energy matching), "QAOA p tuned": QAOA-type circuit with p layers and angles from the Kullback–Leibler objective of Section 4.2, edge-local QITE (5×0.1) or the QITE-limit step (QITE limit). DD+Twirl denotes dynamical decoupling and twirling, and the shots column gives the shots per state. State code: "learned" on π, "learned π^2" on sharpened targets, "transferred" from the learned 10-bit model (on π, or from the π^2-learned one where marked), "fixed": random initialisation kept. Action code "refitted": noise-aware strings from the channel refitted on the runs of the preceding rows.

preparation	action code	state code	S/A/H	shots	CE	top-1	top-2	top-3
QAOA-based	register 8	learned	50/8/60	1024	3.954	0.270	0.479	0.537
QAOA-based	register 10	learned	48/10/60	1024	3.893	0.270	0.472	0.523
QAOA p=1 tuned, DD+Twirl	register 12, noise-aware	learned π^2, full schedule	46/12/60	16384	3.912	0.270	0.359	0.396
QAOA p=3 tuned, DD+Twirl	register 12, noise-aware	learned π^2, full schedule	46/12/60	16384	3.681	0.332	0.472	0.635
QAOA p=5 tuned, DD+Twirl	register 12, noise-aware	learned π^2, full schedule	46/12/60	16384	3.630	0.369	0.582	0.659
QAOA p=3 tuned, DD+Twirl	register 12, refitted	learned π^2, full schedule	46/12/60	16384	3.739	0.270	0.549	0.583
edge-local QITE	register 8	learned	50/8/60	1024	3.910	0.271	0.482	0.555
edge-local QITE, DD+Twirl	register 8	learned	50/8/60	1024	3.837	0.270	0.482	0.544
edge-local QITE 2×0.25	register 8	learned	50/8/60	1024	4.054	0.270	0.479	0.509
edge-local QITE 10×0.05	register 8	learned	50/8/60	1024	3.964	0.276	0.450	0.547
edge-local QITE 20×0.025	register 8	learned	50/8/60	1024	4.028	0.272	0.370	0.472
QITE limit	register 8	learned	50/8/60	1024	3.779	0.359	0.502	0.610
QITE limit, DD+Twirl	register 8	learned	50/8/60	1024	3.686	0.349	0.531	0.637
QITE limit	register 10	learned	48/10/60	1024	3.710	0.373	0.505	0.580
QITE limit, DD+Twirl	register 10	learned	48/10/60	1024	3.711	0.387	0.511	0.594
QITE limit, DD+Twirl	register 10	learned	48/10/60	16384	3.641	0.370	0.496	0.582
QITE limit, DD+Twirl	register 10, plain	fixed	48/10/60	16384	3.698	0.377	0.479	0.570
QITE limit, DD+Twirl	register 10, noise-aware	fixed	48/10/60	16384	3.689	0.419	0.479	0.571
QITE limit, DD+Twirl	register 10, noise-aware	learned	48/10/60	16384	3.619	0.458	0.515	0.631
QITE limit, DD+Twirl	register 12, noise-aware	transferred, π^2	46/12/60	16384	3.620	0.464	0.589	0.691
QITE limit, DD+Twirl	register 14, noise-aware	transferred, π^2	44/14/60	16384	3.624	0.458	0.621	0.688
QITE limit, DD+Twirl	register 16, noise-aware	transferred, π^2	42/16/60	16384	3.597	0.463	0.572	0.681
QITE limit, DD+Twirl	register 16, noise-aware	transferred (π^2-learned), π^2	42/16/60	16384	3.602	0.466	0.558	0.654
QITE limit, DD+Twirl	register 10, noise-aware	learned π^2	48/10/60	16384	3.649	0.484	0.637	0.686
QITE limit, DD+Twirl	register 12, noise-aware	learned π^2, short schedule	46/12/60	16384	3.566	0.488	0.662	0.734
QITE limit, DD+Twirl	register 12, noise-aware	learned π^2, full schedule, run 1	46/12/60	16384	3.555	0.483	0.680	0.750
QITE limit, DD+Twirl	register 12, noise-aware	learned π^2, full schedule, run 2	46/12/60	16384	3.554	0.490	0.675	0.745
QITE limit, DD+Twirl	register 12, noise-aware	learned π^2, full schedule	46/12/60	32768	3.551	0.494	0.680	0.748
QITE limit, DD+Twirl	register 12, refitted	learned π^2, full schedule	46/12/60	16384	3.556	0.485	0.692	0.768
QITE limit, DD+Twirl	register 12, refitted	learned π^2, full schedule	46/12/60	32768	3.555	0.484	0.691	0.768
QITE limit, DD+Twirl	register 14, noise-aware	learned π^2, full schedule	44/14/60	16384	3.563	0.483	0.679	0.746
QITE limit	register 16	fixed	42/16/60	1024	3.787	0.391	0.479	0.565
QITE limit, DD+Twirl	register 16	fixed	42/16/60	1024	3.780	0.382	0.479	0.562
QITE limit, DD+Twirl	register 16	fixed	42/16/60	16384	3.723	0.390	0.479	0.565

first to be sampled on the quantum processor, and the 12-bit full-schedule register models of the last rows of Table 1, with noise-aware strings and, retrained, with strings for the refitted channel. The paragraphs follow the three losses they isolate, the read-out, the temperature of the prepared state and the noise of the quantum processor, and each closes with the decision it led to. None of the findings depends on the grid task: they concern the interface between a Boltzmann machine and a sampler that returns strings, and they transfer to any register model sampled on any lattice.

Single-layer runs Model: the 36-bit grid RBM with the dense random action code, sampled with the single-layer circuit at its energy-matched angles. The two unmitigated sampling runs ran within a minute of submission and used about five seconds of quantum processor time each (304 128 shots per run). The samples are informative about the model but barely about the state: the 36 action-bit means deviate from $\frac{1}{2}$ by 0.10 on average, following the fields of the Hamiltonian, yet their variation across the 297 state codes (standard deviation 0.018) is at the shot-noise floor (0.016). No sample hits a valid action code exactly. The mean Hamming distance to the nearest code is 10.9, the value expected for random 36-bit strings. Decoded to a policy, the samples reach $\mathrm{CE} = 4.35$ and top-1 0.025 with the nearest-code histogram (ties broken at random) and $\mathrm{CE} = 4.64$ and top-1 0.078 with the product-Bernoulli read-out of the bit means (Table 11). A state-independent control that uses the global bit means for every state scores 4.62 and 0.000, so the small top-1 gain is the only trace of the input. Two losses overlap here, a read-out that cannot recover a dense code from samples and

Table 10: Exact read-outs, classical sampling references and noiseless circuit emulations of the grid models (816 validation rows), read out like the quantum-processor samples of Table 1: 1024 samples per state code unless stated, hidden units discarded, action bits decoded to the nearest code (ties at random) or through their means (product-Bernoulli likelihood of the codes). With a register every sampled string is an action. $\beta = 1$ is the model itself, β_{eff} the energy-matched temperature of the single-layer circuit of Section 4.2 of the respective model.

model	action code	S/A/H	edges	fan-in	CE	top-1	top-2	top-3
grid RBM (visible class 0, Table 1), exact softmax read-out	random 36	22/36/60	210	2–4	3.429	0.627	0.746	0.806
grid RBM, visible class 1, exact softmax read-out	random 36	24/36/58	210	2–4	3.425	0.596	0.752	0.838
grid RBM, Metropolis samples, $\beta = 1$, nearest code	random 36	22/36/60	210	2–4	4.080	0.391	0.450	0.507
grid RBM, Metropolis samples, $\beta = 1$, bit means	random 36	22/36/60	210	2–4	4.056	0.395	0.453	0.469
grid RBM, block Gibbs, $\beta = 1$, 8192 samples per state, nearest code	random 36	22/36/60	210	2–4	4.12	0.31		
grid RBM, code-constrained Gibbs, $\beta = 1$, 1 chain $\times$ 1024	random 36	22/36/60	210	2–4	9.2	0.23		
grid RBM, code-constrained Gibbs, $\beta = 1$, 256 chains $\times$ 4	random 36	22/36/60	210	2–4	4.019	0.479	0.626	0.716
grid RBM, Metropolis samples, $\beta_{\text{eff}} = 0.21$, nearest code	random 36	22/36/60	210	2–4	4.363	0.075	0.123	0.127
grid RBM, Metropolis samples, $\beta_{\text{eff}} = 0.21$, bit means	random 36	22/36/60	210	2–4	4.310	0.199	0.219	0.260
grid RBM, 8-bit register, exact read-out	register 8	50/8/60	210	2–4	3.594	0.353	0.520	0.664
grid RBM, 8-bit register, block Gibbs, $\beta = 1$, 1024 per state	register 8	50/8/60	210	2–4	3.713	0.348		
grid RBM, 8-bit register, block Gibbs, $\beta = 1$, 4 chains $\times$ 4096	register 8	50/8/60	210	2–4	3.598	0.353	0.516	0.668
grid RBM, 8-bit register, Metropolis, $\beta = 1$	register 8	50/8/60	210	2–4	3.703	0.353		
grid RBM, 8-bit register, Metropolis, $\beta_{\text{eff}} = 0.24$	register 8	50/8/60	210	2–4	3.801	0.270	0.482	0.538
grid RBM, 8-bit register, noiseless QAOA-based (Farhi et al., 2014; Díez-Valle et al., 2023)	register 8	50/8/60	210	2–4	3.878	0.270		
grid RBM, 8-bit register, noiseless compiled QITE, 2×0.25	register 8	50/8/60	210	2–4	3.946	0.270		
grid RBM, 8-bit register, noiseless compiled QITE, 5×0.1	register 8	50/8/60	210	2–4	3.811	0.270	0.490	0.515
grid RBM, 8-bit register, noiseless compiled QITE, 10×0.05	register 8	50/8/60	210	2–4	3.778	0.272		
grid RBM, 8-bit register, noiseless compiled QITE, 20×0.025	register 8	50/8/60	210	2–4	3.761	0.290		
grid RBM, 8-bit register, fixed code, prior-proportional strings, exact	register 8	50/8/60	210	2–4	3.669	0.277	0.485	0.642
grid RBM, 8-bit register, fixed code, prior-proportional strings, Gibbs	register 8	50/8/60	210	2–4	3.957	0.294	0.438	0.564
grid RBM, 8-bit register, fixed code, equal strings (3–4 per action), exact	register 8	50/8/60	210	2–4	4.152	0.327	0.447	0.534
grid RBM, 8-bit register, fixed code, equal strings (3–4 per action), Gibbs	register 8	50/8/60	210	2–4	4.215	0.191	0.277	0.337
grid RBM, 10-bit register, fixed code, noise-aware, targets π, exact	register 10	48/10/60	210	2–4	3.682	0.424	0.478	0.566
grid RBM, 10-bit register, fixed code, noise-aware, targets π^2, exact	register 10	48/10/60	210	2–4	3.684	0.445	0.545	0.607
grid RBM, 10-bit register, fixed code, noise-aware, targets π^3, exact	register 10	48/10/60	210	2–4	3.727	0.445	0.555	0.657
grid RBM, 10-bit register, fixed code, noise-aware, counts $\propto \sqrt{\pi}$, exact	register 10	48/10/60	210	2–4	3.794	0.380	0.473	0.561
grid RBM, 12-bit register, fixed code, noise-aware, targets π^2, exact	register 12	46/12/60	210	2–4	3.638	0.453	0.613	0.657
grid RBM, 12-bit register, transferred code, noise-aware, π^2, exact	register 12	46/12/60	210	2–4	3.627	0.466	0.602	0.701
grid RBM, 14-bit register, transferred code, noise-aware, π^2, exact	register 14	44/14/60	210	2–4	3.627	0.460	0.638	0.690
grid RBM, 16-bit register, transferred code, noise-aware, π^2, exact	register 16	42/16/60	210	2–4	3.603	0.464	0.589	0.685
grid RBM, 10-bit register, code learned on π^2, noise-aware, exact	register 10	48/10/60	210	2–4	3.669	0.488	0.638	0.683
grid RBM, 12-bit register, π^2-learned code transferred, noise-aware, exact	register 12	46/12/60	210	2–4	3.632	0.480	0.609	0.689
grid RBM, 14-bit register, π^2-learned code transferred, noise-aware, exact	register 14	44/14/60	210	2–4	3.632	0.479	0.600	0.646
grid RBM, 16-bit register, π^2-learned code transferred, noise-aware, exact	register 16	42/16/60	210	2–4	3.603	0.466	0.565	0.651
grid RBM, 12-bit register, code learned on π^2, noise-aware, exact	register 12	46/12/60	210	2–4	3.588	0.494	0.678	0.745
grid RBM, 12-bit register, code learned on π^2, full schedule, noise-aware, exact	register 12	46/12/60	210	2–4	3.583	0.512	0.691	0.749
grid RBM, 12-bit register, code learned on π^2, full schedule, refitted strings, exact	register 12	46/12/60	210	2–4	3.557	0.490	0.705	0.773
grid RBM, 14-bit register, code learned on π^2, full schedule, noise-aware, exact	register 14	44/14/60	210	2–4	3.581	0.494	0.686	0.756
grid RBM, 10-bit register, exact read-out	register 10	48/10/60	210	2–4	3.635	0.369	0.517	0.574
grid RBM, 10-bit register, block Gibbs, $\beta = 1$, 4 chains $\times$ 4096	register 10	48/10/60	210	2–4	3.642	0.395		
grid RBM, 10-bit register, Metropolis, $\beta_{\text{eff}} = 0.23$	register 10	48/10/60	210	2–4	3.86	0.270		
grid RBM, 10-bit register, noise-aware strings, learned code, exact	register 10	48/10/60	210	2–4	3.612	0.458	0.517	0.626
grid RBM, 10-bit register, noise-aware strings, learned code, Gibbs 1024	register 10	48/10/60	210	2–4	3.752	0.440	0.522	0.618
grid RBM, 16-bit register, exact read-out	register 16	42/16/60	210	2–4	3.718	0.390	0.479	0.565
grid RBM, 16-bit register, block Gibbs, $\beta = 1$, 1024 per state	register 16	42/16/60	210	2–4	3.803	0.376	0.482	0.576

Table 11: Runs of the 36-bit grid RBM on `ibm_phoenix` (816 validation rows, 297 distinct state codes, 1024 shots per state unless stated), read out like the rows of Table 1. QAOA-based: the pseudo-Boltzmann circuit of Section 4.2. QITE: the compiled edge-local imaginary-time evolution of Section 4.3.

model	action code	S/A/H	edges	fan-in	CE	top-1	top-2	top-3
grid RBM, QAOA-based (Farhi et al., 2014; Díez-Valle et al., 2023), nearest code	random 36	22/36/60	210	2–4	4.349	0.025	0.042	0.126
grid RBM, QAOA-based (Farhi et al., 2014; Díez-Valle et al., 2023), bit means	random 36	22/36/60	210	2–4	4.641	0.078	0.123	0.191
grid RBM, QAOA-based, twirling + dynamical decoupling, nearest code	random 36	22/36/60	210	2–4	4.377	0.015	0.021	0.028
grid RBM, QAOA-based, expectation values with zero-noise extrapolation, bit means	random 36	22/36/60	210	2–4	4.676	0.004	0.021	0.184
grid RBM, edge-local QITE (Motta et al., 2020), 2×0.25, nearest code	random 36	22/36/60	210	2–4	4.480	0.017	0.042	0.103
grid RBM, edge-local QITE, 2×0.25, bit means	random 36	22/36/60	210	2–4	5.061	0.077	0.110	0.121

a prepared state that is too hot to carry the input, and the paragraphs on read-out and temperature below separate them classically. Decision: no further runs of dense-code models. The action code has to be a register (Section 3.6) and the preparation has to reach the model's own temperature (Section 4.3). The observation transfers to any model whose admissible outputs are a sparse subset of the sampled strings.

Error mitigation Model: the same 36-bit grid RBM and the same 297 circuits. Two further runs test whether mitigation recovers the part lost on the quantum processor. The sampling run with dynamical decoupling (an X, $X^\dagger$ sequence) and Pauli twirling of the CZ layers and of the measurements (32 randomisations, 1024 shots) gives $\mathrm{CE} = 4.38$ / top-1 0.015 with the nearest code and 4.56 / 0.007 with the bit means. The expectation-value run measures $\langle Z_j \rangle$ of the 36 action qubits per state with twirled measurement mitigation, gate twirling and zero-noise extrapolation and dynamical decoupling, 2048 shots per noise factor, and gives 4.68 / 0.004 through the bit-mean read-out. Neither improves on the unmitigated run: the mitigated bit means deviate from $\frac{1}{2}$ by 0.09 on average, as before, but their variation across states (0.017) stays below the reported standard error (0.023). The state dependence that the ideal tempered state carries is a signal of a few hundredths in each bit mean, and that is below what 1024 to 2048 shots resolve on this quantum processor. Mitigation removes bias, not shot noise. The 56 seconds of quantum processor time of the mitigated runs were about eleven times those of a plain run. Decision: dynamical decoupling and twirling were kept as the default of every later run, because they cost quantum processor time but no circuit depth and do remove bias once the signal is large enough to see (in Table 9 mitigation lowers the cross-entropy of the 8-bit register by 0.09 nats and leaves the 10- and 16-bit ones unchanged within shot noise), whereas the expectation-value route with zero-noise extrapolation was dropped: it returns expectation values, and the register read-out needs the joint strings. The general lesson is that mitigation cannot substitute for shots when the per-state signal in a bit mean is below $1/\sqrt{\text{shots}}$, which led to the shot-noise study below.

Imaginary-time evolution Model: the 36-bit grid RBM. The two-step edge-local QITE run (509 CZ gates per circuit) gives 4.48 / 0.017 by nearest code and 5.06 / 0.077 by bit means, with a state dependence of the bit means (0.026) slightly above that of the single-layer run. Decision: edge-local imaginary time on the densely coupled 36-bit model was not pursued further. Its 96-qubit component needs domains that cover the correlations (next paragraph), which no shallow circuit provides.

Noiseless validation of the compiled QITE Models: a 9-qubit patch of the 8-bit register model, a 16-qubit block of the 36-bit grid RBM, and the full conditional model of the 8-bit register. The compiled QITE was validated exactly on these pieces before any run on the quantum processor. On the 9-qubit patch two edge-local steps give fidelity 0.97 with the coherent Gibbs state at $\beta = 1$ and five steps 0.98 (effective β 0.89 and 0.95), and radius-one domains reach fidelity 1.000. The optimum of the single-layer circuit (Section 4.2) on the same patch has fidelity 0.06. The Monte-Carlo-compiled coefficients reproduce the exact-state ones (0.967 vs 0.965). On the densely coupled 16-qubit block of the 36-bit model, however, edge-local QITE with five steps reaches only $\beta_{\text{eff}} = 0.25$ (fidelity 0.33) against 0.21 for the single layer: without larger domains it does not resolve the correlations of that model. For the 8-bit register model the coupling graph of the free qubits splits into 36 components of at most five qubits, so its noiseless QITE policy can be computed exactly. Fidelity is a misleading guide here: the components reach 0.91 to 0.99, but the fidelity is carried by the strongly biased hidden qubits, while the single action qubit of a component, whose marginal is close to $\frac{1}{2}$, is off by up to 0.2, and the register read-out amplifies exactly these errors. The noiseless QITE policy is 3.95 / 0.270 with two steps, 3.81 / 0.270 with five, 3.78 / 0.272 with ten and 3.76 / 0.290 with twenty, against 3.63 / 0.355 for 2048 exact samples per state drawn component by component, a sampling estimate of the exact read-out 3.594 / 0.353, and 3.88 / 0.270 for the noiseless single-layer circuit: imaginary time lowers the cross-entropy steadily but converges slowly in the marginals that matter. On `ibm_phoenix` (53 seconds of quantum processor time for three runs) the 8-bit register model gives 4.05 / 0.270 with two steps (128 CZ) and 3.91 / 0.271 with five (320 CZ), and the 36-bit model with two steps (509 CZ) stays below top-1 0.1 like its single-layer runs (Table 11). Longer edge-local schedules do not help on the quantum processor: ten steps of 0.05 (641 CZ, two-qubit depth 81) give 3.96 / 0.276 and twenty steps of 0.025 (1288 CZ, depth 162) 4.03 / 0.272, the growing circuit noise outweighing the small noiseless gain. Error mitigation does not change this: the five-step run with dynamical decoupling and twirling gives 3.84 / 0.270, closer to its noiseless value of 3.81 but still at the prior, because the noiseless edge-local state is itself at the prior. Three decisions follow. Every circuit is judged by the policy it produces, not by its fidelity, because the read-out weights the qubits differently from the fidelity. The edge-local schedules were abandoned in favour of the QITE-limit step of Section 4.3, which is exact for the star components at a fraction of the depth. And the star structure itself, one action qubit with private hidden neighbours, was kept as a design constraint of every later register model, since it is

what makes the exact step cheap. The exactness argument holds for any model whose conditional components are trees or small, independent of the task.

Single-layer runs of the registers Models: the 8-bit and the 10-bit register models with the learned state code. The 8-bit register (50 state, 8 action and 60 hidden units, 256 strings, action 6 holding 40 of them) reaches CE $= 3.594$ and top-1 0.353 with the exact read-out, the capacity price of the 28 dead units and the shorter schedule. Its samples converge to the exact read-out, and the prior-proportional assignment of the strings is the better of the two assignments we compared (paragraph on the string assignment below). The single-layer circuit for this Hamiltonian has the energy-matched angles $\gamma_M = 0.149$, $\theta_M = 1.024$ and $\beta_{\text{eff}} = 0.24$, and at that temperature the ideal conditional is already the prior (paragraph on the temperature below). The quantum-processor run (564 distinct validation states, 64 CZ gates per circuit, 1024 shots, 20 seconds of quantum processor time) returns 3.95 / 0.270 / 0.479 / 0.537, the state-independent prior with some noise. The 10-bit register (48 state units, 1024 strings, 20 dead hidden units) behaves the same way: exact read-out 3.635 / 0.369 / 0.517 / 0.574, $\beta_{\text{eff}} = 0.23$ at $\gamma_M = 0.157$, $\theta_M = 1.081$, and the quantum-processor run (547 states, 80 CZ gates per circuit) 3.89 / 0.270 / 0.472 / 0.523. The register therefore removes the read-out loss completely, and what remains is the temperature of the one-layer state. Decision: the register was kept as the action code of every later model, and the effort moved from the read-out to the state preparation. The tuned single layers of the 12-bit models (below) later confirmed that no choice of angles changes this.

Placement sweep Model: the 10-bit register with the fixed random state code, trained by closed form and fine-tuning only. Eight alternative placements of the ten action vertices (a separate seed for the placement, the same split), each with 20 dead hidden units, reach 3.70–3.72 / 0.34–0.36 against 3.69 / 0.38 for the default placement in the same mode. The placement changes the result by at most 0.03 nats, less than the learned state code or the string assignment. Decision: the default placement (the placement of seed 0, which every later model shares) was frozen, and the search effort went into the state code and the string assignment, whose effects are about twice as large. The finding transfers to other lattices with defects: as long as every action vertex keeps its private hidden neighbours, where exactly the register sits on the lattice is a second-order choice.

Estimation of the flip channel The noise-aware string assignment of Section 3.6 and the predicted quantum-processor policies of Section 5.2 rest on a per-qubit asymmetric bit-flip channel of the quantum processor, fitted from a QITE-limit run. For an action qubit q and a validation state s the exact marginal $m_q(s) = p(\sigma_q = +1 \mid s)$ follows by enumeration of the tree component that contains q (at most 13 qubits), and the measured marginal $\hat{m}_q(s)$ is the mean of the qubit's readings over the shots. A channel that flips a 1 to 0 with probability p_{10} and a 0 to 1 with probability p_{01} maps the one onto the other,

$$\hat{m}_q(s) = (1 - p_{10})\, m_q(s) + p_{01}\,(1 - m_q(s)), \tag{10}$$

so (p_{10}, p_{01}) are the least-squares solution of (10) over all validation states, clipped to $[0, 1]$. The fit absorbs read-out and state-preparation errors of that run alike. Its residual is 0.003 to 0.009 per qubit, so the linear model describes the data. Pushing the exact string distribution of a register model through the product of these channels predicts its quantum-processor policy (Section 4.4). The flip weights $-\log f_q$ with $f_q = (p_{10} + p_{01})/2$ enter the noise-aware assignment. The channel fitted on 16 September 2026 on the 10-bit register found one qubit flipping in $31\,\%$ of the shots (qubit 15) and all others below $4\,\%$, with qubit 28 at $14\,\%$. The channel fitted six days later on the 12-bit full-schedule register (action qubits 8, 11, 15, 33, 48, 51, 55, 73, 77, 91, 95 and 108) found qubit 15 at $3\,\%$ and instead qubit 55 at $31\,\%$, qubit 77 at $11\,\%$ and qubit 95 at $8\,\%$, the remaining qubits between 1 and $6\,\%$. Two runs three hours apart gave the same channel to within a percentage point. Because the string assignment of that model was built for the earlier channel, it protected the wrong qubit, which is the loss discussed in Section 5.2. A third fit, on the run of the retrained model one day later, gave the same ranking (qubit 55 at $25\,\%$, 77 at $13\,\%$, 73 and 95 at $8\,\%$). Three decisions rest on this channel. The noise-aware string assignment of Section 3.6 puts the strings of the frequent actions where a flip of the worst qubit does not change the action. The predicted quantum-processor policy is computed before every run and compared with the measurement afterwards, which is how the systematic loss of the 12-bit model was recognised as a moved qubit rather than shot noise. And because the channel drifts within days, the strings of the final model were re-derived from a channel

fitted immediately before its training, after which the quantum processor reproduced its exact read-out within shot noise. The fit needs nothing but a run with an exact reference marginal, so it transfers to any register model and, with the integrated control errors in place of the flip rates, to the annealers of Appendix G.

Tuned QAOA-type circuits Models: the 12-bit full-schedule register with noise-aware strings and the retrained one with strings for the refitted channel. For these models every conditional component is a star of at most five qubits, so the QAOA-type circuit of Section 4.2 can be emulated exactly. We generalise the single layer to p layers, $R_y(\theta)U_1(\lambda)\, e^{-i\gamma_p \hat{E}} \prod_{k=1}^{p-1} \left[R_x(2\phi_k)^{\otimes n} e^{-i\gamma_k \hat{E}}\right]$ applied to $|+\rangle^{\otimes n}$ (time-ordered, $k = 1$ acting first), with the single-qubit gates on every free qubit, and choose the $2p + 1$ angles $\gamma_1, \ldots, \gamma_p$, $\phi_1, \ldots, \phi_{p-1}$ (mixers between the layers only, so $p = 1$ is exactly the circuit of Section 4.2), θ and λ by minimising the mean Kullback–Leibler divergence $\mathrm{KL}(p_{\mathrm{Gibbs}} \,\|\, p_{\mathrm{circuit}})$ of the component distributions over the 1455 distinct training states (Nelder–Mead from several starts, warm-started across depths). No teacher label enters the objective. Table 12 reports the exact validation policy of the tuned circuits and the runs on the quantum processor. We make two observations. First, the single layer is at the prior for every choice of angles: the energy-matching angles and the KL-tuned angles give the same top-1 of 0.270, so the failure of the single layer in Table 1 is a property of the ansatz, not of its parameters. Second, depth buys agreement slowly: the divergence falls from 0.33 to 0.03 nats per state between one and five layers, top-1 rises from 0.27 to 0.39 in emulation and to 0.37 on the quantum processor, whereas the QITE-limit step reaches the exact read-out (0.512) noiselessly and 0.490 to 0.494 on the quantum processor with the two-qubit gate count of the single layer. The quantum processor reproduces every tuned circuit to within a few hundredths of a nat. For the model with the refitted strings even three layers leave top-1 at the prior. The gain appears in top-2 and top-3 only. Decision: the QAOA-type preparation was not pursued further, and the QITE-limit step remains the preparation of every final result in Table 1. The tuning procedure itself, exact emulation of the components and a divergence objective on the training states, applies to any model whose conditional components are small, and it is the right way to compare approximate preparations before spending quantum processor time on them.

Table 12: Tuned QAOA-type circuits for the two 12-bit models: mean $\mathrm{KL}(p_{\mathrm{Gibbs}} \| p_{\mathrm{circuit}})$ per validation state (angles tuned on the training states), controlled-Z gates of the compiled circuit, exact validation policy of the ideal circuit, and the run on the quantum processor (16384 shots, dynamical decoupling and twirling).

model	angles	KL	CZ	ideal CE / top-1/2/3	QPU CE / top-1/2/3
noise-aware	energy matching, 1 layer	0.360	94	3.919 / 0.270 / 0.359 / 0.386	–
noise-aware	KL-tuned, 1 layer	0.326	94	3.956 / 0.270 / 0.359 / 0.386	3.912 / 0.270 / 0.359 / 0.396
noise-aware	KL-tuned, 2 layers	0.133	188	3.747 / 0.270 / 0.436 / 0.612	–
noise-aware	KL-tuned, 3 layers	0.085	282	3.714 / 0.331 / 0.423 / 0.609	3.681 / 0.332 / 0.472 / 0.635
noise-aware	KL-tuned, 4 layers	0.045	376	3.657 / 0.387 / 0.471 / 0.642	–
noise-aware	KL-tuned, 5 layers	0.031	470	3.641 / 0.392 / 0.506 / 0.647	3.630 / 0.369 / 0.582 / 0.659
noise-aware	QITE-limit step	0	94	3.583 / 0.512 / 0.691 / 0.749	3.554 / 0.490 / 0.675 / 0.745
refitted	energy matching, 1 layer	0.298	94	3.872 / 0.270 / 0.301 / 0.511	–
refitted	KL-tuned, 1 layer	0.274	94	3.870 / 0.270 / 0.479 / 0.511	–
refitted	KL-tuned, 2 layers	0.107	188	3.744 / 0.270 / 0.540 / 0.565	–
refitted	KL-tuned, 3 layers	0.070	282	3.742 / 0.271 / 0.561 / 0.585	3.739 / 0.270 / 0.549 / 0.583
refitted	QITE-limit step	0	94	3.557 / 0.490 / 0.705 / 0.773	3.556 / 0.485 / 0.692 / 0.768

Classical students of comparable size Models: two classical students, trained on the same 3735 training rows with the teacher's action distribution as soft target and evaluated on the same 816 validation rows. A multinomial logistic regression on the 20 standardised line loadings (1722 parameters) reaches $\mathrm{CE} = 3.294$ / top-1/2/3 0.621 / 0.797 / 0.871, a two-layer perceptron with 28 hidden units and a hyperbolic tangent (2966 parameters, the parameter count of the dense RBM) 3.133 / 0.786 / 0.924 / 0.950, and one with 128 hidden units (13 266 parameters) 3.080 / 0.850 / 0.951 / 0.973. Weight decay was chosen by validation cross-entropy among four values, Adam for 3000 full-batch steps. The dense RBM of Table 1 (3.193 / 0.703) therefore sits between the linear model and the small perceptron, and the grid RBM (3.429 / 0.627) at the linear model's level. This is the expected order: the RBM's binary state code and bipartite structure are constraints that the perceptron does not carry, and they are the price of a model whose Gibbs state a quantum processor

can prepare. Decision: none for the pipeline, but the numbers fix the scale of the ladder in Table 1. A reader should judge the grid RBM as a sampleable student of linear-model quality, not as a state-of-the-art distillation.

Standard errors Models: every policy we could recompute, i.e. the exact read-outs of the register models and of the grid RBM (Tables 10 and 12) and every quantum-processor row of Table 1. The 816 validation rows come from 20 scenarios whose rows are correlated (consecutive states of one grid trajectory), so a binomial standard error over rows (0.017 for a top-1 near 0.5) understates the uncertainty. We resample the 20 scenarios with replacement 2000 times and recompute CE and top-k on the resampled rows. The resulting standard errors are 0.047 to 0.060 nats for CE, 0.026 to 0.046 for top-1, 0.033 to 0.052 for top-2 and 0.035 to 0.059 for top-3, with no systematic dependence on the model: the mean policy has 3.781 ± 0.047 / 0.270 ± 0.039, the exact 36-bit grid RBM 3.429 ± 0.050 / 0.627 ± 0.039, the exact 12-bit full-schedule model 3.583 ± 0.055 / 0.512 ± 0.045 and its retrained version 3.557 ± 0.055 / 0.490 ± 0.045. Seed variation is of the same size: the code learning of the dense model moves top-1 by up to six points between seeds (Table 6), and the two full-schedule 12-bit models of Table 10, which differ only in their string assignment, reach 0.512 and 0.490 exactly. The steps of the register ladder in Section 5.2 (two to four top-1 points each) are therefore individually within one standard error. What the ladder establishes is the monotone trend and its total, from 0.35 to 0.49, which is three standard errors, and the agreement between a quantum-processor policy and its exact read-out, which is a paired comparison on the same states with a much smaller error (the runs repeat to within 0.01). Decision: the text of Section 5.2 states the direction and size of every step but claims significance only for the total and for the exact-versus-quantum-processor comparisons.

Unconditional samples on every qubit and coupler Models: the 36-bit grid RBM and the 12-bit full-schedule register model, sampled without clamping. In the conditional runs the clamped state qubits turn their couplers into fields, so two-qubit gates act on the 47 to 127 hidden–action couplers only. To sample the joint distribution of all 118 units we ran both families of preparations on the unconditional Ising model, with a two-qubit gate on every one of the 210 couplers and every functioning qubit measured: the single-layer circuit at its energy-matched angles (420 controlled-Z gates, two-qubit depth 8) and two edge-local imaginary-time steps of $\Delta\tau = 0.25$ compiled for the unconditional model (841 gates, one of the 420 two-qubit unitaries compiling to three gates instead of two, depth 16), each with dynamical decoupling and twirling and $65\,536$ to $131\,072$ shots. There is no exact read-out of a 118-qubit joint state, so the check is statistical (Table 13). The mean energy of the samples defines an effective inverse temperature β_{fit} by matching the energy curve of block-Gibbs samples of the model. Its shot-noise uncertainty is below 0.001. The samples are then compared with block-Gibbs samples at β_{fit} through their 118 magnetisations $\langle\sigma_i\rangle$, their 210 connected nearest-neighbour correlations $\langle\sigma_i\sigma_j\rangle - \langle\sigma_i\rangle\langle\sigma_j\rangle$, which vanish for any product state and therefore test whether the couplers act, and the spread of their energy, which a Boltzmann state at β_{fit} fixes as well. A fields-only control, the product state with the couplings zeroed at the same β_{fit}, is listed for comparison, and two independent block-Gibbs samples of the model agree with each other to 0.998 in these statistics. For the single layer the samples follow the tempered model in every statistic: β_{fit} is $86\,\%$ of the ideal value (0.18 against 0.21 for the grid RBM, 0.13 against 0.15 for the register model), the magnetisations agree with correlation 0.91, the connected correlations with 0.96 and 0.94 at a root-mean-square deviation of 0.05 and a mean magnitude of 0.12 against 0.11 in the model, and the energy spread agrees to $3\,\%$ and $1\,\%$. The fields-only control matches the magnetisations even better (0.98), as expected at a fifth of the model's inverse temperature, where the fields dominate the first moments, but it has no connected correlations and correlates with the raw pair correlations at only 0.18 and 0.27: the second moments, not the first, are the evidence that every coupler takes part. What the single layer produces is a tempered version of the model, five to eight times hotter than the model itself at the measured effective temperatures (four to seven at the ideal ones of Section 4.2), not the model. The two edge-local steps, which are noiseless-exact only for $\Delta\tau \to 0$ and carry twice the gates, reach $\beta_{\mathrm{fit}} = 0.10$ and 0.07, their connected correlations agree with the tempered model at only 0.43 and 0.27, and their energy spread exceeds the Boltzmann value by 11 and $16\,\%$: at 841 two-qubit gates on 118 qubits the noise of the quantum processor dominates. Decision: these runs establish that every functioning qubit and coupler of the quantum processor takes part in sampling a tempered version of the trained model, which the conditional runs alone did not show, and they confirm on the full 118-qubit state the finding of the conditional runs that

the single layer is a well-formed but too hot Boltzmann sampler and that edge-local imaginary time does not reach the model's temperature on this quantum hardware.

Table 13: Unconditional sampling of the whole model on `ibm_phoenix`, all 118 qubits measured and two-qubit gates on all 210 couplers (dynamical decoupling and twirling). β_{fit}: inverse temperature whose block-Gibbs mean energy equals the measured one (ideal single-layer value in brackets). r: Pearson correlation between the measured and the block-Gibbs values at β_{fit} of the 118 magnetisations m, the 210 raw pair correlations c and the 210 connected correlations c^{conn}, with the root-mean-square deviation (RMSE) of the latter. σ_E: energy spread of the samples against that of the block-Gibbs samples. "fields only": the product state with the couplings zeroed at the same β_{fit}, scored against the same measurement.

model	preparation	CZ	shots	β_{fit}	r_m	r_c	$r_{c^{\mathrm{conn}}}$ / RMSE	σ_E measured / model	fields only: r_m, r_c
grid RBM (36-bit)	single layer	420	131k	0.18 (0.21)	0.91	0.91	0.96 / 0.05	19.0 / 18.4	0.98, 0.18
grid RBM (36-bit)	edge-local QITE 2×0.25	841	66k	0.10	0.71	0.44	0.43 / 0.16	24.9 / 22.5	0.68, 0.31
register 12	single layer	420	131k	0.13 (0.15)	0.91	0.91	0.94 / 0.05	24.8 / 24.6	0.97, 0.27
register 12	edge-local QITE 2×0.25	841	66k	0.07	0.56	0.32	0.27 / 0.17	32.5 / 28.0	0.50, 0.26

Joint samples of action and hidden units Models: the 12-bit full-schedule register model and its retrained version, re-analysing the shots of their 32768-shot QITE-limit runs. The conditional state is a product of stars, and in each star the marginal of the action qubit is set by the rotation on the centre, so the policy alone does not show that the two-qubit gates do anything but add noise. The measured samples do show it: for every validation state and every star we compare the connected correlation $\langle\sigma_c\sigma_l\rangle - \langle\sigma_c\rangle\langle\sigma_l\rangle$ between the action qubit c and each hidden leaf l with its exact value in the conditional Gibbs state. Over the 22 231 (full-schedule) and 20 304 (retrained) action–hidden pairs the measured values follow the exact ones with Pearson correlation 0.983 and 0.998, root-mean-square deviation 0.070 and 0.031 and a regression slope of 0.82 and 0.92, the shrinkage expected from the flip channel. Half of the exact connected correlations exceed 0.1 in magnitude (mean magnitude 0.19, measured 0.17), whereas any product of single-qubit rotations would give zero. The hidden-leaf marginals agree with correlation 0.998 and 0.996. The same shots also explain why four quantum-processor rows of Table 1 have a lower cross-entropy than their exact read-out: the flip channel and the pseudo-count of $1/82$ per action flatten the model's over-confident action distribution slightly, and the teacher's own distribution is broad (3.02 nats of entropy), so the flattening costs top-k but lowers the cross-entropy by up to 0.03 nats. Decision: the claim that the quantum processor reads out joint samples of the action and hidden units rests on these correlations, not on the policy. The check needs only the stored shots and the exact star distributions and applies to any model whose conditional components are small.

Shot-noise floor Models: the 10-bit and the 16-bit register models (exact string distributions). Ideal multinomial samples of the exact string distribution of the 10-bit register give 3.705 / 0.401 with 1024 samples per state, 3.646 with 4096 and 3.638 with 16384, against 3.635 exactly. For the 16-bit register 3.773, 3.729 and 3.721 against 3.718. The mitigated 1024-shot quantum-processor runs sit on these floors: their gap to the exact read-out is what a noiseless sampler with the same number of shots would show, so a 1024-shot run cannot be compared with an exact read-out at the level of a few hundredths of a nat. Decision: every run of Table 1 from the 10-bit register on uses 16384 shots per state, the last ones 32768, and the floor is computed for a model before its shot budget is chosen. The floor depends only on the entropy of the model's string distribution and the number of shots, so the same computation applies to any register model on any sampler, classical or quantum.

Read-out Model: the 36-bit grid RBM with the dense random action code. Metropolis samples at $\beta = 1$, i.e. samples of the RBM itself, give $\mathrm{CE} = 4.06$ and top-1 0.395 with the bit-mean read-out and 4.08 / 0.391 with the nearest code, far below the 3.43 / 0.627 of the softmax read-out. This is not a matter of sample size: the softmax marginalises the hidden units analytically and normalises over the 82 admissible codes, whereas a sample of the 36 action bits almost never is an admissible code (none of $3 \cdot 10^5$ samples was), so nearest-code decoding estimates the mass of each code's Voronoi cell and the bit-mean read-out a product approximation. With 8192 unconstrained block-Gibbs samples per state the nearest-code policy still stands at 4.12 / 0.31. A sampler that respects the constraint, drawing the action as one of the 82 codes given the hidden units, does converge to

the softmax policy, but the RBM's conditionals are so sharp ($p(y \mid h)$ has 0.46 nats of entropy, its mode carries 0.83 of the mass) that a single chain locks into a mode: 1024 samples from one chain give CE $= 9.2$ / top-1 0.23, while 256 independent chains with 4 samples each give 4.02 / 0.48 / 0.63 / 0.72, approaching the analytical values. This is the loss that the classical sampling row of Table 1 shows and that no quantum processor can undo, because it is a property of the code, not of the sampler. Decision: dense codes were given up for sampling in favour of the fully assigned register (Section 3.6), in which every string is an action and the sample-based read-out converges to the exact one (paragraph on registers below). The argument transfers to any model whose output code occupies a vanishing fraction of the sampled strings. A quantum sampler, which returns one joint sample per measurement and cannot be constrained to admissible codes, is affected by it more than a classical chain that can.

Temperature Models: the 36-bit grid RBM and the 8- and 10-bit registers, sampled classically at the effective temperature of the single-layer circuit. At the energy-matched $\beta_{\text{eff}} = 0.21$ the unconstrained read-outs of the 36-bit model give 4.31 / 0.199 (bit means) and 4.36 / 0.075 (nearest code). For the register models the tempered conditional is already the prior: Metropolis samples at $\beta_{\text{eff}} = 0.24$ (8-bit) and 0.23 (10-bit) give 3.80 / 0.270 and 3.86 / 0.270, the baseline values. These classical numbers reproduce the quantum-processor runs of the single layer (Table 9), so the runs failed for a classical reason: at a fifth of the model's inverse temperature the conditional of the action bits is flat. Decision: the preparation has to reach $\beta = 1$ exactly, which is what the QITE-limit step of Section 4.3 does. A tempered approximation is not an option for a policy whose information sits in the sharpness of the conditional. The check itself, sampling the model classically at the circuit's effective temperature before the run, transfers to any approximate Gibbs-state preparation.

Registers Models: the 8-, 10- and 16-bit register models. With a fully assigned register every sampled string is an action and the samples converge to the exact read-out. For the 8-bit register, block Gibbs at $\beta = 1$ gives 3.713 / 0.348 from 1024 samples per state and 3.598 / 0.353 / 0.516 / 0.668 from four chains of 4096, with a mean absolute deviation of 0.0007 from the exact conditional. Metropolis sampling gives 3.703 / 0.353. The 10-bit register converges the same way (3.642 / 0.395 from four chains of 4096, mean deviation 0.0009), and the 16-bit register gives 3.80 / 0.376 from 1024 samples per state against 3.718 / 0.390 exactly. Decision: this classical convergence check became the pre-flight of every model before its run on the quantum processor, since it separates a failure of the code from a failure of the preparation. A register model that does not pass it is not sent to the quantum processor (registers that duplicate every bit on three qubits and decode by majority vote, which we also trained, were dropped for this reason: their exact read-out is competitive, but their block-Gibbs samples fall to the prior because the copies are not tied given the hidden units). The check applies to any model with a fully assigned output register.

String assignment Model: the 8-bit register with the fixed random state code, trained with the same seed under two assignments. The extra strings of a register are handed to the actions in proportion to the prior, so that a uniform string distribution maps to the prior and the model only encodes the deviation from it. The price is that action 6 holds 40 of 256 strings and wins wherever the bits are not sharp. The alternative, 3 to 4 strings per action, gains exact top-1 (0.327 against 0.277) but its cross-entropy (4.15) is worse than the prior, because the model has to carry the prior itself with marginals it cannot make sharp enough, and 1024 samples no longer resolve the arg-max among three-string actions (4.22 / 0.191). A flattened string distribution would map to the uniform policy (CE $= 4.41$) instead of the prior, so the equal assignment is also the worse failure mode on the quantum processor. Decision: every later register uses the prior-proportional assignment, on top of which the noise-aware assignment of Section 3.6 decides which strings go to which action. The principle transfers to any register: the extra strings should carry the prior of the output, so that a sampler that has lost the input degrades to the prior rather than to the uniform distribution.

Sharpened targets, string counts and code transfer Models: the 10-bit register with the fixed random state code and the noise-aware strings (targets and string counts), the 12-, 14- and 16-bit registers with the state code transferred from the learned 10-bit model (width), and the 12-bit register with its own learned code (schedules). With the fixed random state code and the noise-aware assignment, training the 10-bit register on π^2 raises top-1 from 0.424 to 0.445 and top-2 from 0.478 to 0.545 at unchanged cross-entropy. π^3 adds top-2 and top-3 at a cost of 0.045 nats. String

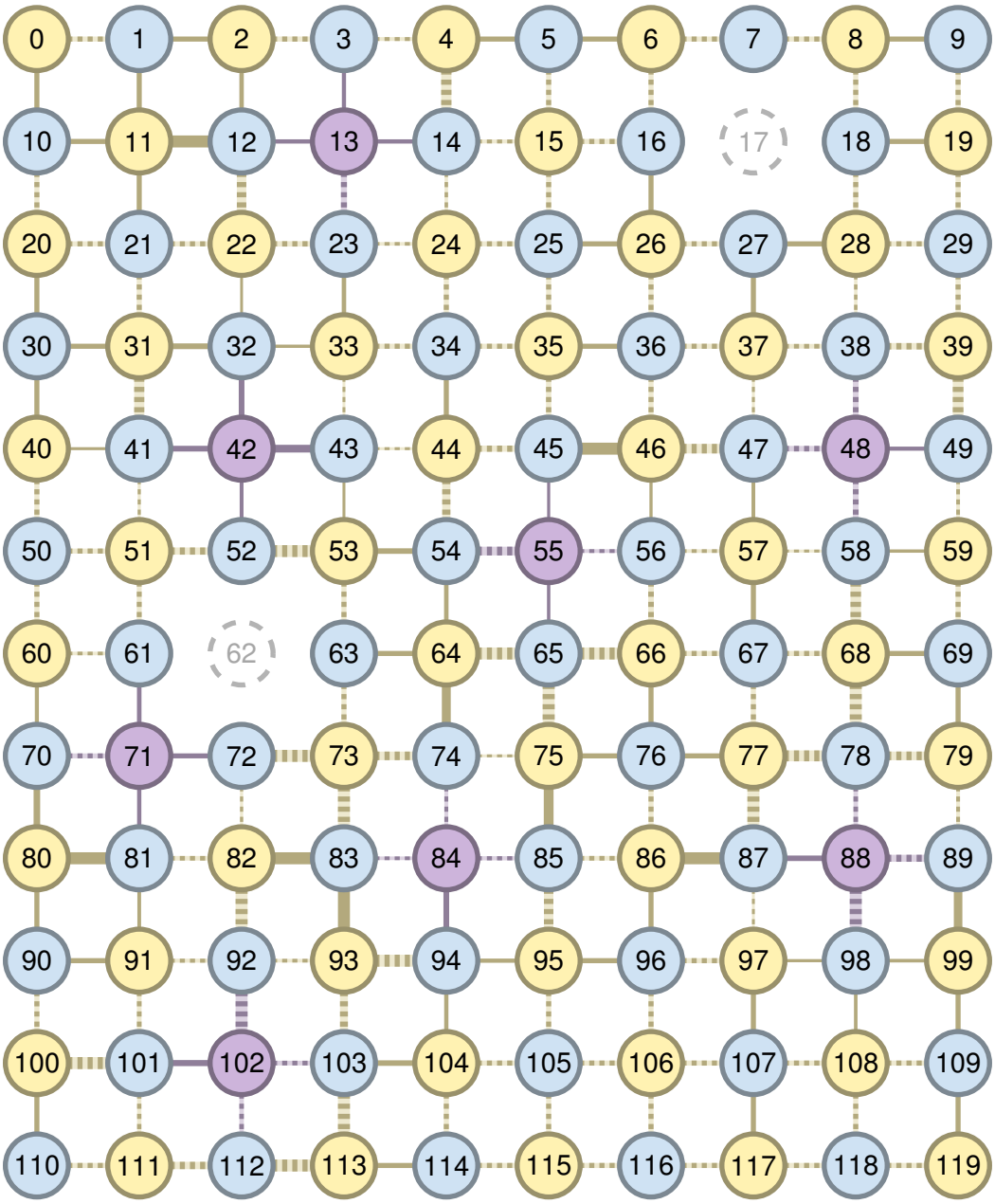

Figure 3: Grid embedding of the 8-bit register RBM (50 state, 8 action and 60 hidden units), drawn as Fig. 2: vertices numbered row by row from 0 to 119, defects 17 and 62 dashed, hidden units blue, state bits yellow, register bits lavender. Line width by weight magnitude (saturating at $|W| = 6$), light/dark pattern for negative weights (129 of 210 edges). Weights range from -12.5 to $+10.2$. Median magnitude 1.5, mean 2.2, 19 below 0.5. The 28 hidden units without a lavender edge are dead for the read-out. Each register bit with its four hidden neighbours is one star component of the conditional model.

counts proportional to $\sqrt{\pi}$ instead of π hurt every metric (3.794 / 0.380). A 12-bit register with the random code and π^2 reaches 3.638 / 0.453. With the state code transferred from the learned 10-bit model the 12-, 14- and 16-bit registers reach 3.627 / 0.466, 3.627 / 0.460 and 3.603 / 0.464, so the transferred code is worth about 0.01 in cross-entropy and top-1 over the random code, and register width beyond 12 bits improves cross-entropy and top-2 but not top-1. Ensembles of the register models do not help: averaging the best two policies gives 0.466, averaging more lowers the

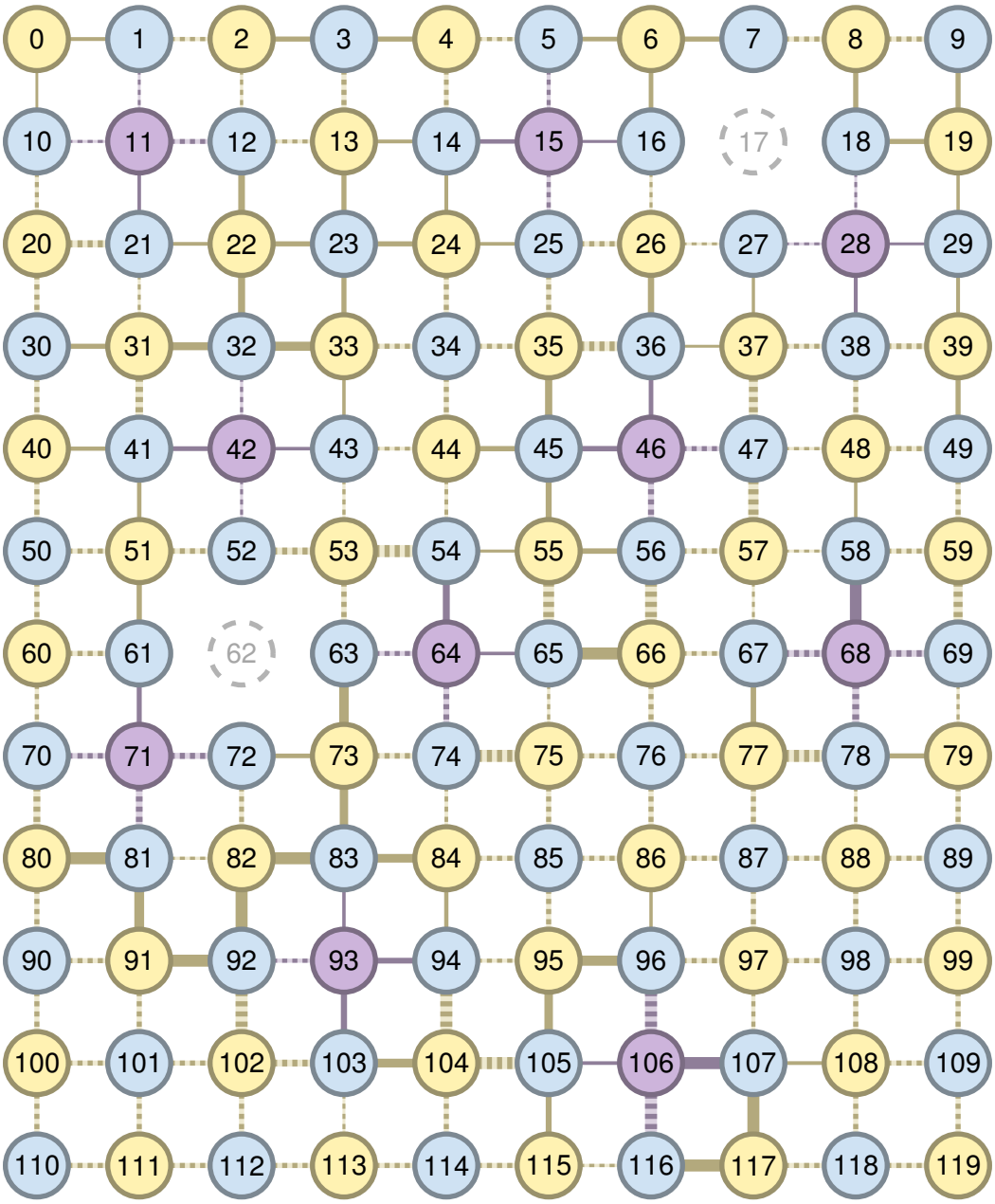

Figure 4: Grid embedding of the 10-bit register RBM (48 state, 10 action and 60 hidden units), drawn as Fig. 2: vertices numbered row by row from 0 to 119, defects 17 and 62 dashed, hidden units blue, state bits yellow, register bits lavender. Line width by weight magnitude (saturating at $|W| = 6$), light/dark pattern for negative weights (132 of 210 edges). Weights range from -11.7 to $+13.3$. Median magnitude 1.5, mean 2.1, 33 below 0.5. The 20 hidden units without a lavender edge are dead for the read-out. Each register bit with its four hidden neighbours is one star component of the conditional model.

agreement. The 12-bit register with its own state code learned on π^2 reaches 3.588 / 0.494 exactly with a shortened schedule and 3.583 / 0.512 with the full schedule (Section 5.2). 1024 block-Gibbs samples per state read the former out at 3.760 / 0.502. Decisions: the target π^2 was adopted for every later model and π^3 and the $\sqrt{\pi}$ counts were dropped. The code transfer served as a cheap probe of the register width, whose outcome, that width buys calibration and not the arg-max, fixed 12 bits as the register of the final models. Ensembles were dropped. And the state code is learned

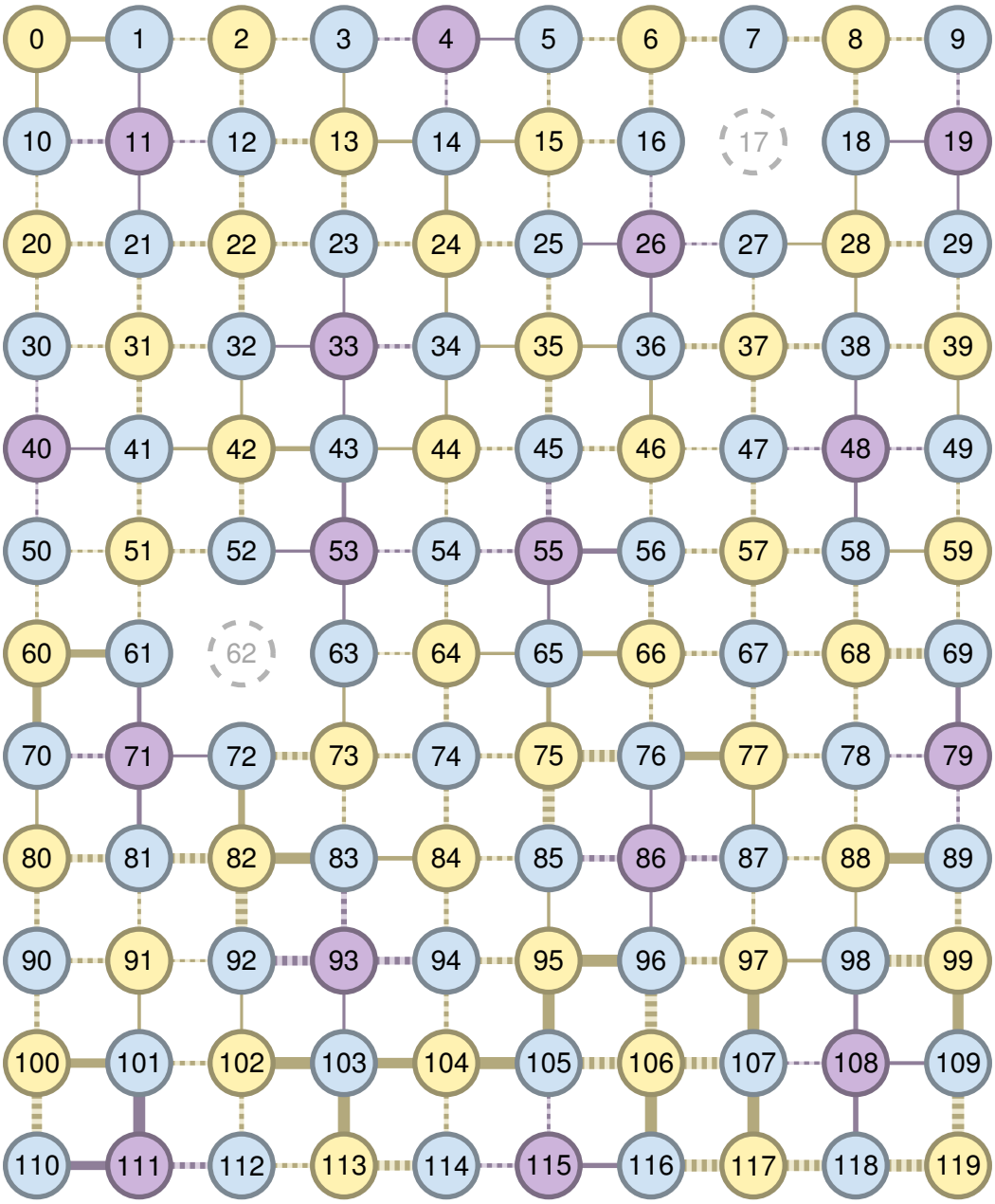

Figure 5: Grid embedding of the 16-bit register RBM (42 state, 16 action and 60 hidden units, fixed random state code), drawn as Fig. 2. Line width by weight magnitude (saturating at $|W| = 6$), light/dark pattern for negative weights (130 of 210 edges). Weights range from -10.8 to $+13.3$. Median magnitude 1.2, mean 1.8, 68 below 0.5. Only 4 hidden units lack a lavender edge. The free-qubit components are all trees: thirteen with one register bit, and one of 13 qubits (vertices 23, 32, 33, 34, 43, 45, 52, 53, 54, 55, 56, 63, 65) with three register bits linked through the hidden units 43 and 54.

for each register with the full schedule, which lifts the exact top-1 of the 12-bit register above one half. The sharpened target and the width finding are properties of distillation into a register and not of the power-grid task: any teacher with a peaked action distribution and a student read out through the arg-max benefits from the former, and any register wider than the code needs pays the latter.

F THE QITE-LIMIT CIRCUIT

Fig. 6 shows the circuit of Section 4.3 for one star component of a register model: an action qubit c and its hidden neighbours $l_1, \ldots, l_4$ on the lattice, all starting in $|0\rangle$. Clamping the state qubits turns their couplings into the fields f_i of the free qubits, so every angle is a classical function of the clamped state and one circuit is compiled per distinct validation state code (473 for the 12-bit models). The rotation on the centre encodes the marginal of the action qubit with the leaves summed out, $\cos^2(\theta/2) = p(\sigma_c = +1)$. The uniformly controlled rotation on leaf l encodes the two conditionals $\cos^2(\varphi_l^{\pm}/2) = p(\sigma_l = +1 \mid \sigma_c = \pm 1)$ through $a_l = (\varphi_l^+ + \varphi_l^-)/2$ and $b_l = (\varphi_l^+ - \varphi_l^-)/2$: with the control in $|0\rangle$ ($\sigma_c = +1$) the two controlled-NOT gates act trivially and the leaf receives $R_y(a_l + b_l) = R_y(\varphi_l^+)$, with the control in $|1\rangle$ the second rotation is conjugated to $R_y(-b_l)$ and the leaf receives $R_y(\varphi_l^-)$. The prepared state is $\sqrt{p(\sigma_c)}\prod_l \sqrt{p(\sigma_l \mid \sigma_c)}$, the coherent Gibbs state of the star at $\beta = 1$. A statevector check against the exact distribution agrees to 10^{-15}. Hidden qubits without an action neighbour are singletons and receive one R_y each. The full circuit of a 12-bit model is the product of eleven five-qubit stars, one four-qubit star at a lattice boundary and 13 singletons: compiled to the native gate set of `ibm_phoenix` (controlled-Z, $\sqrt{X}$ and R_z) with the identity layout it has 94 controlled-Z gates, two-qubit depth 8 and total depth 32, followed by one measurement of the 72 free qubits.

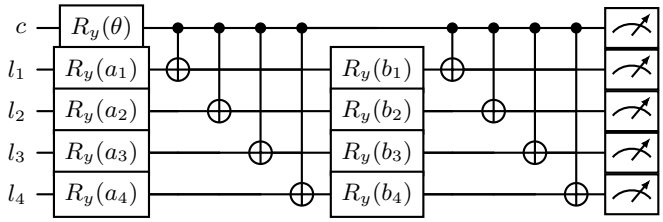

Figure 6: QITE-limit imaginary-time step for one star component: action qubit c (centre) and its four hidden neighbours $l_1, \ldots, l_4$ (leaves), all starting in $|0\rangle$. One R_y on the centre, one uniformly controlled R_y (two controlled-NOT gates) per leaf, and one measurement. The angles θ, a_l, b_l are given in the text. The eight controlled-NOT gates all act between lattice neighbours, so no routing is needed.

G BOLTZMANN MACHINES SAMPLED ON QUANTUM HARDWARE

Table 14 lists the published experiments we are aware of that sample a Boltzmann machine on quantum hardware, with the model size and the machine. All large models run on quantum annealers, and the samples follow a Boltzmann distribution at an effective temperature that has to be estimated from the data (Benedetti et al., 2016). An annealer is an analogue machine without a compiled circuit. The gate-based experiments prepare or sample Gibbs states of a few qubits. The quantum-enhanced Markov chain of Layden et al. (2023) uses a 10-qubit circuit as the proposal of a classical chain for spin glasses, not for a trained model. Our 118-qubit model is the largest RBM on a gate-based quantum processor, and the only one in the table whose lattice embedding needs no chains: every unit is one physical qubit and every weight one coupler.

Chains on annealers and depth on gate-based quantum processors A quantum annealer has the same limitation as the gate-based quantum processor of Section 1: each qubit couples to few neighbours, six on the Chimera lattice of the D-Wave 2000Q and its predecessors (Bunyk et al., 2014), fifteen on the Pegasus lattice of the Advantage systems (Boothby et al., 2020) and twenty on the Zephyr lattice of the Advantage2 generation (Boothby et al., 2021). An annealer has no SWAP gates, so a model whose graph does not fit the lattice is minor-embedded (Choi, 2008): every logical unit becomes a chain of physical qubits held together by strong ferromagnetic couplers, so that the chain acts as one variable and its members can reach the couplers of all its neighbours. A fully connected RBM of $64 + 64$ units on the 2000Q takes chains of 16 qubits (Dixit et al., 2021). The chain plays the role of the SWAP sequence, and it fails in the same way: the couplers and fields of an annealer carry integrated control errors, small random deviations of the programmed values (D-

Wave Systems, 2024), and the longer the chain, the more of them accumulate and the higher the probability that the chain breaks and its qubits disagree (D-Wave Systems, 2025; Jeong et al., 2025), just as a longer SWAP sequence accumulates gate errors and decoherence on a gate-based quantum processor. Chain breaks have to be repaired by majority vote or the sample discarded, and the chain strength has to grow with the chain length, which compresses the range left for the model's own couplings (Choi, 2008; Pochart et al., 2021). The methods of this paper therefore transfer to annealers: a model trained directly on the Chimera, Pegasus or Zephyr lattice with one qubit per unit needs no chains, its couplings use the full programmable range, and the noise-aware action code can be built from the measured integrated control errors of the qubits that carry the register in the same way as from the flip channel of Section 4.4.

Table 14: Boltzmann machines sampled on quantum hardware. All D-Wave machines are quantum annealers. Units: visible + hidden (logical) units of the largest model. "chains": logical units embedded as chains of physical qubits.

work	quantum hardware	model	units
Adachi & Henderson (2015)	D-Wave Vesuvius (512 qubits, annealer)	RBM layers of a deep belief network, chains	$32 + 32$
Benedetti et al. (2017)	D-Wave 2X	sparse model embedded in the qubit graph, 940 qubits	46 logical
Caldeira et al. (2020)	D-Wave 2000Q	RBM, chains	$48 + 48$
Job & Adachi (2020)	D-Wave	RBM layers of a deep belief network, chains	$32 + 32$
Dixit et al. (2021)	D-Wave 2000Q	RBM, 16-qubit chains	$64 + 64$
Dixit et al. (2022)	D-Wave 2000Q	RBM on 64-bit data, chains	64+hidden
Higham & Bedford (2023)	D-Wave Advantage	energy-based model from a trained neural network	358 nodes
Kālis et al. (2023)	D-Wave	RBM, hybrid inference	–
Salmenperä & Nurminen (2023)	D-Wave Advantage	RBM, unit dropout at $256 + 256$	$64 + 64$, $256 + 256$
Moro & Prati (2023)	D-Wave	RBM for anomaly detection	–
Sinno et al. (2025)	D-Wave Advantage	RBM, chains	$120 + 120$
Salloum et al. (2025)	D-Wave Advantage 1, Advantage 2 prototype	RBM on MNIST	–
El Yazizi et al. (2025)	D-Wave Advantage 4.1	RBM, chains	$74 + 74$, $154 + 72$
El-Yazizi & Koshka (2025)	D-Wave Advantage	RBM, continual learning	–
Wu et al. (2020)	IBM `ibmq_essex` (gate-based)	Gibbs state of a Haldane chain	4 qubits
Layden et al. (2023)	IBM (gate-based)	quantum-enhanced Markov chain Monte Carlo for Ising spin glasses, no trained model	10 qubits
this work	IBM `ibm_phoenix` (gate-based)	conditional RBM, one qubit per unit	$58 + 60$ (118 qubits)

H IMPLEMENTATION

All model-side operations are implemented natively over the sufficient statistics of the model. The RBM is represented as a Markov random field over binary vertices whose pairwise cliques join one visible and one hidden vertex. Each pairwise block has four overcomplete entries, three of which are locked at zero so that a block is the single weight W_{vh} and an absent connection is an all-zero block. The loss object stores one role per vertex (input, target, hidden), the admissible codes with their action map, and the training rows grouped by input state. It provides the objective and gradient of (3), prediction, the input gradient for the binariser, the closed form with both selection rules, the fan-in penalty with its proximal operator and projection (including the per-group minima), and a real-valued input mode for the relaxed forward pass. The learned binariser is an optimisation problem over (V, c) solved with the same optimisers as the model (gradient descent, FISTA with and without backtracking, L-BFGS). The alternation between model and binariser only refreshes statistics between calls. Gradients are factorised over target-activation tables and grouped soft inputs, which made the relaxed rounds four times faster. The quantum circuits are built from the stored model: the Ising coefficients, the single-layer angles, the compiled imaginary-time unitaries and the tree preparations are all computed from the model file, transpiled to the quantum processor with the identity layout, and read out with the model's code-to-action map.