Quantum-enhanced Markov chain Monte Carlo

David Layden,[1,*] Guglielmo Mazzola,[2] Ryan V. Mishmash,[1] Mario Motta,[1] Pawel Wocjan,[3] Jin-Sung Kim,[1,†] and Sarah Sheldon[1]

[1] IBM Quantum, Almaden Research Center, San Jose, California 95120, USA
[2] IBM Quantum, IBM Research – Zurich, 8803 Rüschlikon, Switzerland
[3] IBM Quantum, T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

Sampling from complicated probability distributions is a hard computational problem arising in many fields, including statistical physics, optimization, and machine learning. Quantum computers have recently been used to sample from complicated distributions that are hard to sample from classically, but which seldom arise in applications. Here we introduce a quantum algorithm to sample from distributions that pose a bottleneck in several applications, which we implement on a superconducting quantum processor. The algorithm performs Markov chain Monte Carlo (MCMC), a popular iterative sampling technique, to sample from the Boltzmann distribution of classical Ising models. In each step, the quantum processor explores the model in superposition to propose a random move, which is then accepted or rejected by a classical computer and returned to the quantum processor, ensuring convergence to the desired Boltzmann distribution. We find that this quantum algorithm converges in fewer iterations than common classical MCMC alternatives on relevant problem instances, both in simulations and experiments. It therefore opens a new path for quantum computers to solve useful—not merely difficult—problems in the near term.

MCMC

Quantum computers promise to solve certain computational problems much faster than classical computers. However, current quantum processors are limited by their modest size and appreciable error rates. Recent efforts to demonstrate quantum speedups have therefore focused on problems that are both classically hard and naturally suited to current quantum devices, like sampling from complicated—though not explicitly useful—probability distributions [1–3]. Here we introduce and experimentally demonstrate a quantum algorithm that is similarly well-suited to current devices, which samples from distributions that can be both complicated and useful: the Boltzmann distribution of classical Ising models.

A classical Ising model consists of n variables $(s_1, \ldots, s_n) = \boldsymbol{s}$ called spins that can take values $s_j = \pm 1$ independently [4]. A model instance is defined by coefficients $\{J_{jk}\}_{j>k=1}^{n}$ and $\{h_j\}_{j=1}^{n}$ called couplings and fields respectively, together with a temperature $T > 0$. It is often represented as a graph, as in Figs. 1a-b. Each spin configuration $\boldsymbol{s} \in \{1, -1\}^n$ is assigned an energy

$$\boxed{E(\boldsymbol{s}) = -\sum_{j>k=1}^{n} J_{jk} s_j s_k - \sum_{j=1}^{n} h_j s_j} \tag{1}$$

and a corresponding Boltzmann probability $\mu(\boldsymbol{s}) = \frac{1}{\mathcal{Z}} e^{-E(\boldsymbol{s})/T}$, where the partition function $\mathcal{Z} = \sum_{\boldsymbol{s}} e^{-E(\boldsymbol{s})/T}$ ensures normalization. Sampling from μ is a common subroutine in many disparate applications, including in statistical physics, where it is used to compute thermal averages [5]; in machine learning, to train Boltzmann machines [6]; and in combinatorial optimization, as part of the famous simulated annealing algorithm [7]. Indeed, this sampling is often a computational bottleneck when J_{jk} and h_j have varying signs and follow no particular pattern. Eq. (1) typically defines a rugged energy landscape for such instances, informally termed *spin glasses* [8], with many local minima that can be far from one another in Hamming distance, as depicted in Fig. 1c. In the $T \to 0$ limit, sampling from μ amounts to minimizing $E(\boldsymbol{s})$, which is NP-hard for spin glasses [9]. For small but non-zero T, sampling from μ requires finding several of the lowest-energy configurations, which may be far apart, not just the ground configuration(s). This hard regime will be our main focus.

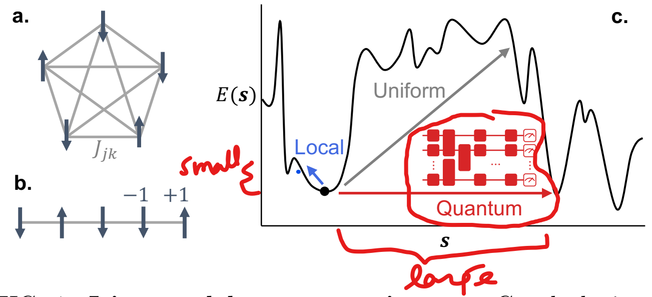


FIG. 1. **Ising model representations. a.** Graph depicting an $n = 5$ model instance where arrows (vertices) represent spins and edges represent the $\binom{n}{2}$ non-zero couplings J_{jk}. Fields h_j are not shown. **b.** An $n = 5$ model instance with only $n - 1$ non-zero couplings. **c.** A rugged energy landscape typical of spin glasses, with the configurations $\boldsymbol{s} \in \{-1, 1\}^n$ depicted in 1D. Typical proposed jumps for three MCMC algorithms, from a local minimum, are shown for illustration.

MARKOV CHAIN MONTE CARLO

Markov chain Monte Carlo (MCMC) is the most popular algorithmic technique for sampling from the Boltzmann distribution μ of Ising models in all of the aforementioned applications. In fact, it was named one of the ten most influential algorithms of the 20th century in science and engineering [10]. It approaches the problem indirectly, without ever computing $\mu(\boldsymbol{s})$ and in turn the partition function $\mathcal{Z}$, which is believed to take $\Omega(2^n)$ time. Rather, MCMC performs a sequence of random jumps between spin configurations, starting from a generic one and jumping from $\boldsymbol{s}$ to $\boldsymbol{s'}$ with fixed transition probability $P(\boldsymbol{s'}|\,\boldsymbol{s})$ in any iteration. Such a process is called a Markov chain. For it to form a useful algorithm, the transitions probabilities must be carefully chosen to make the process converge to μ; that is, for the probability of being in $\boldsymbol{s}$ after many iterations to approach $\mu(\boldsymbol{s})$. Sufficient conditions for such convergence are that the Markov chain be irreducible and aperiodic (technical requirements that will always be met here [11]), and satisfy the detailed balance condition:

$$P(\boldsymbol{s'}|\,\boldsymbol{s})\,\mu(\boldsymbol{s}) = P(\boldsymbol{s}\,|\boldsymbol{s'})\,\mu(\boldsymbol{s'}) \tag{2}$$

for all $\boldsymbol{s} \neq \boldsymbol{s'}$ [12].

How to efficiently realize a P satisfying these conditions? A powerful approach is to decompose each jump into two steps: *Step 1 (proposal)*- If the current configuration is $\boldsymbol{s}$, propose $\boldsymbol{s'}$ with some probability $Q(\boldsymbol{s'}|\,\boldsymbol{s})$. *Step 2 (accept/reject)*- Compute an appropriate acceptance probability $A(\boldsymbol{s'}|\,\boldsymbol{s})$ based on Q and μ, then move to $\boldsymbol{s'}$ with that probability (i.e., accept the proposal), otherwise remain at $\boldsymbol{s}$. This gives $P(\boldsymbol{s'}|\,\boldsymbol{s}) = A(\boldsymbol{s'}|\,\boldsymbol{s})\,Q(\boldsymbol{s'}|\,\boldsymbol{s})$ for all $\boldsymbol{s'} \neq \boldsymbol{s}$. For a given Q there are infinitely many choices of A that satisfy detailed balance (2). One of the most popular is the Metropolis-Hastings (M-H) acceptance probability [13, 14]:

$$A(\boldsymbol{s'}|\,\boldsymbol{s}) = \min\left(1,\ \frac{\mu(\boldsymbol{s'})}{\mu(\boldsymbol{s})}\frac{Q(\boldsymbol{s}\,|\boldsymbol{s'})}{Q(\boldsymbol{s'}|\,\boldsymbol{s})}\right). \tag{3}$$

Although $\mu(\boldsymbol{s})$ and $\mu(\boldsymbol{s'})$ cannot be efficiently computed, Eq. (3) depends only on their ratio, which can be evaluated in $O(n^2)$ time since $\mathcal{Z}$ cancels out. This cancellation underpins MCMC, and ensures that $A(\boldsymbol{s'}|\,\boldsymbol{s})$ can be efficiently computed provided $Q(\boldsymbol{s}\,|\boldsymbol{s'})/Q(\boldsymbol{s'}|\,\boldsymbol{s})$ can too. The same is true for the other popular choice of A, called the Glauber or Gibbs sampler acceptance probability [11].

The most common way to propose a candidate $\boldsymbol{s'}$ is by flipping ($s_j \mapsto -s_j$) a uniformly random spin $j \in [1, n]$ of the current configuration $\boldsymbol{s}$. We call this the *local proposal strategy* since $\boldsymbol{s}$ and $\boldsymbol{s'}$ are always neighbors in terms of Hamming distance. The resulting acceptance probability can be computed efficiently, and the overall Markov chain converges quickly to μ for simple model instances at moderate T [12]. Various non-local strategies can also be used (sometimes in combination [15]), including the *uniform proposal strategy* where $\boldsymbol{s'}$ is picked uniformly at random from $\{-1, 1\}^n$, and more complex ones which flip clusters of spins [16–19]. Broadly, spin glasses at low T present a formidable challenge for all of these approaches, manifesting in long autocorrelation, slow convergence, and ultimately long MCMC running times. This challenge is an active area of research due to the many applications in which it arises [20]. The problem: $\mu(\boldsymbol{s'})/\mu(\boldsymbol{s})$ in Eq. (3), which comes from demanding detailed balance (2), is exponentially small in $\Delta E/T = [E(\boldsymbol{s'}) - E(\boldsymbol{s})]/T$ for energy-increasing proposals $\boldsymbol{s} \to \boldsymbol{s'}$. Consequently, such proposals are frequently rejected. (And a rejected proposal still counts as an MCMC iteration.) This makes it difficult for Markov chains to explore rugged energy landscapes at low T, as they tend to get stuck in local minima for long stretches and to rarely cross large barriers in the landscape.

QUANTUM ALGORITHM

To alleviate this issue, we introduce an MCMC algorithm which uses a quantum computer to propose moves and a classical computer to accept/reject them. It alternates between two steps: *Step 1 (quantum proposal)*- If the current configuration is $\boldsymbol{s}$, prepare the computational basis state $|\boldsymbol{s}\rangle$ on the quantum processor, where $s_j = \pm 1$ refers to an eigenvalue of Z_j. (E.g., if $\boldsymbol{s} = (1, 1, -1)$, prepare $|\boldsymbol{s}\rangle = |001\rangle$. We use X_j, Y_j and Z_j to denote σ_x, σ_y and σ_z on qubit j respectively.) Then apply a unitary U satisfying the symmetry constraint:

$$\big|\langle \boldsymbol{s'}|U|\boldsymbol{s}\rangle\big| = \big|\langle \boldsymbol{s}|U|\boldsymbol{s'}\rangle\big| \ \text{ for all } \boldsymbol{s}, \boldsymbol{s'} \in \{-1, 1\}^n. \tag{4}$$

Finally, measure each qubit in the Z eigenbasis, i.e., the computational basis, denoting the outcome $\boldsymbol{s'}$. *Step 2 (classical accept/reject)*- Compute $A(\boldsymbol{s'}|\,\boldsymbol{s})$ from Eq. (3) on a classical computer and jump to $\boldsymbol{s'}$ with this probability, otherwise stay at $\boldsymbol{s}$. While computing $Q(\boldsymbol{s'}|\,\boldsymbol{s})$ and $Q(\boldsymbol{s}\,|\boldsymbol{s'})$ may take exponential (in n) time in general, there is no need to do so: Eq. (3) depends only on their ratio, which equals 1 since $Q(\boldsymbol{s'}|\,\boldsymbol{s}) = |\langle \boldsymbol{s'}|U|\boldsymbol{s}\rangle|^2 = Q(\boldsymbol{s}\,|\boldsymbol{s'})$ due to (4). This cancellation underpins our algorithm, and mirrors that between $\mu(\boldsymbol{s})$ and $\mu(\boldsymbol{s'})$. The resulting Markov chain provably converges to the Boltzmann distribution μ, but is hard to mimic classically, provided it is classically hard to sample the measurement outcomes of $U|\boldsymbol{s}\rangle$. This combination opens the possibility of a useful quantum advantage.

The symmetry requirement (4) ensures convergence to μ, but does not uniquely specify U. Rather, we pick the quantum step of our algorithm heuristically with the aim of accelerating MCMC convergence in spin glasses at low T, while still satisfying condition (4). We then evaluate the resulting Markov chains through simulations and experiments. Several choices of U are promising, including ones arising from quantum phase estimation and quantum annealing [11]. However, we focus here on evolution $U = e^{-iHt}$ under a *time-independent* Hamiltonian

$$H = (1-\gamma)\alpha\, H_{\text{prob}} + \gamma H_{\text{mix}} \tag{5}$$

for a time t, where

$$H_{\text{prob}} = -\sum_{j>k=1}^{n} J_{jk} Z_j Z_k - \sum_{j=1}^{n} h_j Z_j = \sum_{\boldsymbol{s}} E(\boldsymbol{s}) |\boldsymbol{s}\rangle\langle\boldsymbol{s}| \tag{6}$$

encodes the classical model instance, H_{mix} (discussed below) generates quantum transitions, and $\gamma \in [0,1]$ is a parameter controlling the relative weights of both terms. It is convenient to include a normalizing factor $\alpha = \|H_{\text{mix}}\|_{\text{F}}/\|H_{\text{prob}}\|_{\text{F}}$ in Eq. (5) so that both terms of H share a common scale regardless of $\{J_{jk}, h_j\}$, which can be arbitrary. ($\|M\|_{\text{F}} = \text{tr}(M^\dagger M)^{1/2}$ is the Frobenius norm of a matrix M.)

In principle, H_{mix} could comprise arbitrary weighted sums and products of X_j and $Y_j Y_k$ terms. These produce a symmetric H and therefore $U = U^T$ (although $U \neq U^\dagger$ generically), thus satisfying condition (4). If H_{mix} were a dense matrix and $\gamma \ll 1$, a perturbative analysis shows that non-trivial quantum proposals $\boldsymbol{s} \to \boldsymbol{s}'$ would exhibit a remarkable combination of features that suggest fast MCMC convergence [11]: Like local proposals, their absolute energy change $|\Delta E| = |E(\boldsymbol{s}') - E(\boldsymbol{s})|$ is typically small, meaning they are likely to be accepted even if $\Delta E > 0$. However, like uniform or cluster proposals, $\boldsymbol{s}$ and $\boldsymbol{s}'$ are typically far in Hamming distance, so the resulting Markov chain can move rapidly between distant local energy minima. While a dense H_{mix} would pose experimental difficulties, similar behavior is reported for various spin glasses in Refs. [21–23] using

$$H_{\text{mix}} = \sum_{j=1}^{n} X_j \tag{7}$$

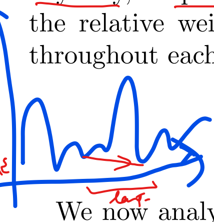
and larger γ. Inspired by these results, we take Eq. (7) as the definition of H_{mix} here. The normalizing factor α in Eq. (5) then takes the simple form

$$\alpha = \frac{\|H_{\text{mix}}\|_{\text{F}}}{\|H_{\text{prob}}\|_{\text{F}}} = \frac{\sqrt{n}}{\sqrt{\sum_{j>k=1}^{n} J_{jk}^2 + \sum_{j=1}^{n} h_j^2}} \tag{8}$$

which—crucially—can be computed in $O(n^2)$ time. Step 1 of our algorithm therefore consists of realizing quenched dynamics of a transverse-field quantum Ising model encoding the classical model instance, which can be efficiently simulated on a quantum computer [24]. Note, however, that our algorithm samples from a classical Boltzmann distribution μ, not from a quantum Gibbs state as in Refs. [25–28].

As an initial state $|\boldsymbol{s}\rangle$ evolves under H, it becomes delocalized due to H_{mix} and effectively queries the classical energy function E from Eq. (1) in quantum superposition through H_{prob}. The measurement outcome $\boldsymbol{s}'$ is therefore influenced by the entire energy landscape. The details of this quantum evolution depend on the free parameters γ and t. Rather than try to optimize these, we rely on a simple observation: instead of applying the same U in every iteration of our algorithm, one could equally well pick U at random in each iteration from an ensemble $\{U_1, U_2, \dots\}$ of unitaries, each satisfying condition (4). This amounts to sampling outputs from random circuits, initialized according to the state of a Markov chain. It is this randomized approach that we implement, by picking γ and t uniformly at random in each iteration [11], thus obviating the need to optimize these parameters. Note that we do not invoke adiabatic quantum evolution in any way, despite the familiar form of Eq. (5). Rather, the relative weights of H_{prob} and H_{mix} are held fixed throughout each quantum step.

Imp



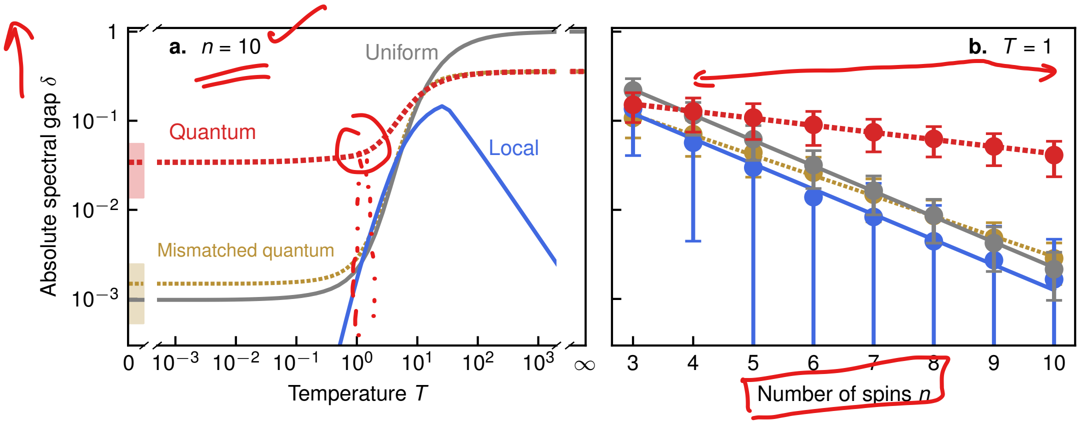
PERFORMANCE

We now analyze the running time of our quantum algorithm through its convergence rate. MCMC convergence is inherently multifaceted. However, a Markov chain's convergence rate is often summarized by its absolute spectral gap, $\delta \in [0,1]$, where the extremes of $\delta = 0$ and 1 describe the slowest and fastest possible convergence, respectively. This quantity is found by forming the transition probabilities $P(\boldsymbol{s}'|\,\boldsymbol{s})$ into a $2^n \times 2^n$ matrix and computing its eigenvalues $\{\lambda\}$, all of which satisfy $|\lambda| \leq 1$ and describe a facet of the chain's convergence. The absolute spectral gap, $\delta = 1 - \max_{\lambda \neq 1} |\lambda|$, describes the slowest facet. More concretely, it bounds the mixing time τ_ε, defined as the minimum number of iterations required for the Markov chain's distribution to get within any $\varepsilon > 0$ of μ in total variation distance, for any initial distribution, as [12]

$$\left(\delta^{-1} - 1\right) \ln\left(\frac{1}{2\varepsilon}\right) \;\leq\; \tau_\varepsilon \;\leq\; \delta^{-1} \ln\left(\frac{1}{\varepsilon \, \min_{\boldsymbol{s}} \mu(\boldsymbol{s})}\right). \tag{9}$$

Since δ is the only quantity on either side of (9) that depends on the proposal strategy Q, it is a particularly good metric for comparing the convergence rate of our quantum algorithm with that of common classical alternatives. Because δ depends on the model instance and on T, not just on Q, we analyzed our algorithm's convergence in two complementary ways: First, we simulated it on a classical computer for many instances to elucidate the average-case δ. Second, we implemented it experimentally for illustrative instances and analyzed both the convergence rate and the mechanism underlying the speedup observed in simulations. (Note that computing δ is different—and much more demanding—than simply running MCMC.) We used the M-H acceptance probability (3) throughout, although this choice has little impact on the results [11]. Unlike previously proposed algorithms which prepare a quantum state encoding μ, ours uses simple, shallow quantum circuits [29–35]. Its alternating quantum/classical structure means that quantum

Converge Rate!



c. $\langle\delta\rangle \propto 2^{-kn}$ **fits**	
Quantum	$k = 0.264(4)$
Mismatched quantum	$k = 0.76(1)$
Local	$k = 0.94(4)$
Uniform	$k = 0.948(7)$
Quantum enhancement factor in k: 3.6(1)	

FIG. 2. **Average-case convergence rate simulations.** The absolute spectral gap δ, a measure of MCMC convergence rate, using the M-H acceptance probability (3) with different proposal strategies. All strategies were simulated classically. Lines/markers show the average δ over 500 random fully-connected Ising model instances for each n; error bands/bars show the standard deviation in δ over these instances. Dotted lines are for visibility. **a.** The slow-down of each strategy at low T. For the local proposal strategy, $\delta \to 0$ also at high T because an eigenvalue of its transition matrix approaches -1. This artifact can be easily be remedied by using a lazy chain or the Gibbs sampler acceptance probability; the same is not true at low T, however [11]. **b.** Problem size dependence, with least squares exponential fits to the average δ, weighted by the standard error of the mean. **c.** The resulting fit parameters and the average quantum enhancement exponent, which is the ratio of k for the quantum algorithm and the smallest k among classical proposal strategies (the local strategy, here). Uncertainties are from the fit covariance matrices. Similar data for different n and T, model connectivity, and acceptance probabilities is shown in [11].

coherence need only be maintained in each repetition of Step 1, rather than over the whole algorithm [36, 37]. Moreover, it generates numerous samples per run, like a fully classical MCMC algorithm, rather than a single one. These features make it sufficiently well-suited to current quantum processors that we observed a quantum speedup experimentally on up to $n = 10$ qubits.

Average-Case Performance

For the first part of the analysis, we generated 500 random spin glass instances on n spins by drawing each J_{jk} and h_j independently from standard normal distributions. We did not explicitly fix any couplings J_{jk} to zero; the random instances are therefore fully connected as in Fig. 1a. This ensemble is the archetypal Sherrington-Kirkpatrick model [38] (up to a scale factor) with random local fields, where the fields serve to break inversion symmetry and thus increase the complexity. For each instance, we explicitly computed all the transition probabilities $\{P(\boldsymbol{s}'|\,\boldsymbol{s})\}$ and then δ as a function of T for different proposal strategies Q. We then averaged δ over the model instances, and repeated this process for $3 \leq n \leq 10$. The results describe the average MCMC convergence rate as a function of n and T. Two illustrative slices are shown in Fig. 2, where n and T are held fixed in turn. At high T, where μ is nearly uniform, the uniform proposal produces a fast-converging Markov chain with δ near 1, as shown in Fig. 2a. However, both the uniform and local proposals suffer a sharp decrease in δ at lower T. This slow-down is much less pronounced for our quantum algorithm, which gives a substantially better δ on average than either classical alternative for $T \lesssim 1$.

For all three proposal strategies, the average scaling of δ with n at $T = 1$ fits well to $\langle\delta\rangle \propto 2^{-kn}$, as shown in Fig. 2b. Our algorithm appears to give an average-case polynomial enhancement in δ over the local and uniform proposals based on the fitted values of k, shown in Fig. 2c. These values depend on T, but their ratios suggest a roughly cubic/quartic enhancement at low temperatures regardless of the exact T [11]. Finally, to elucidate the source of this average-case quantum enhancement, we also computed δ for a *mismatched quantum proposal strategy*, where moves are proposed like in our algorithm, but based on the wrong energy landscape. That is, rather than explore the relevant E defined by coefficients $\{J_{jk}\}$ and $\{h_j\}$ as in Eqs. (1) and (6), the mismatched quantum strategy uses the wrong coefficients $\{\tilde{J}_{jk}\}$ and $\{\tilde{h}_j\}$ (drawn randomly from the same distribution as J_{jk} and h_j) in its quantum Hamiltonian and therefore explores the wrong energy landscape $\tilde{E}$ in Step 1. The resulting δ is comparable with that of the uniform proposal in Fig. 2. This suggests that the δ enhancement in our algorithm indeed arises from exploring the classical energy landscape E in quantum superposition to propose moves.

Experimental Implementation

For the second part of the analysis we focus in on individual model instances, for which we implemented our quantum algorithm experimentally and analyzed it

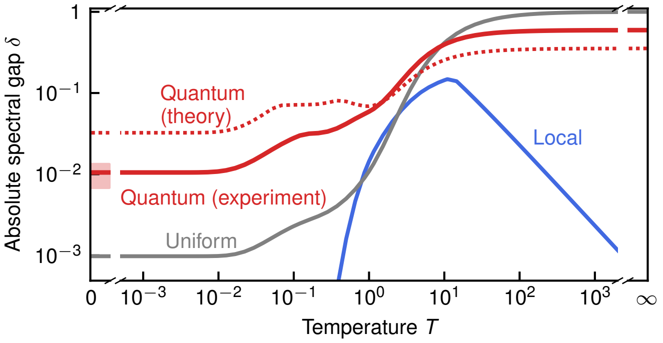


FIG. 3. **Convergence rate experiment.** The absolute spectral gap δ, a measure of MCMC convergence rate, for an illustrative model instance on $n = 10$ spins with 1D connectivity. Each proposal strategy is combined with the M-H acceptance probability (3). To infer δ for the experimental realization of our algorithm, we recorded 5.76×10^7 quantum transitions $|\boldsymbol{s}\rangle \to |\boldsymbol{s}'\rangle$ between computational states with $|\boldsymbol{s}\rangle$ uniformly distributed, to estimate each $Q(\boldsymbol{s}' \,|\, \boldsymbol{s})$. For each T, we used the full data set to estimate the MCMC transition matrix P and in turn form a point estimate of δ (solid red line), then a random subsample of the data to compute 99% bootstrap confidence intervals (red error bands).

in more depth than is feasible for a large number of instances. We generated random model instances and picked illustrative ones whose lowest-E configurations include several near-degenerate local minima—a central feature of spin glasses at larger n which hampers MCMC at low T. We then implemented our quantum algorithm experimentally for these instances on *ibmq_mumbai*, a 27-qubit superconducting quantum processor, using Qiskit [39], an open-source quantum software development platform. (The Ising model T has no relation to the processor's physical temperature.) We approximated $U = e^{-iHt}$ on this device through a randomly-compiled second-order Trotter-Suzuki product formula [24, 40, 41] with up to 48 layers of pulse-efficient 2-qubit gates [42] acting on up to 5 pairs of qubits in parallel [11], and set $Q(\boldsymbol{s} \,|\, \boldsymbol{s}')/Q(\boldsymbol{s}' \,|\, \boldsymbol{s}) = 1$ in the acceptance probability. Unlike in the first part of the analysis, we restricted our focus to model instances where $J_{jk} = 0$ for $|j-k| \neq 1$ as in Fig. 1b, in order to match the connectivity of qubits in the quantum processor for this initial demonstration. Simulations show that the average-case δ for such instances is qualitatively similar to Fig. 2 [11].

We focus on an $n = 10$ model instance here in which the six lowest-E configurations are all local minima, and the two lowest have an energy difference of just $|\Delta E| = 0.05$. Similar instances with $n = 8$ and 9 are analyzed in [11]. For the present $n = 10$ instance, δ closely follows the average in Fig. 2a for the local and uniform proposal strategies, as well as for our quantum algorithm in theory, as shown in Fig. 3. We estimated δ as a function of T for the experimental realization of our quantum algorithm by counting quantum transitions $|\boldsymbol{s}\rangle \to |\boldsymbol{s}'\rangle$ between all computational states $|\boldsymbol{s}\rangle$ and $|\boldsymbol{s}'\rangle$ to estimate the MCMC transition matrix, which we then diagonalized. At low T, the inferred δ is smaller than the theoretical value due to experimental imperfections, but still significantly larger than that of either the local or uniform alternative. This constitutes an experimental quantum enhancement in MCMC convergence on current quantum hardware. At high T, the experimental δ is larger than the theoretical value for our quantum algorithm, which we attribute to noise in the quantum processor mimicking the uniform proposal to a degree. We also plot δ for common MCMC cluster algorithms (those of Swendsen-Wang, Wolff and Houdayer) in [11]. These are substantially more complicated than the local and uniform proposals, both conceptually and practically, but offer almost no δ enhancement in this setting compared to the simpler classical alternatives, so we do not focus on them here.

To further illustrate the increased convergence rate of our quantum-enhanced MCMC algorithm compared to these classical alternatives, we use it to estimate the average magnetization (with respect to the Boltzmann distribution μ) of this same $n = 10$ instance. The magnetization of a spin configuration $\boldsymbol{s}$ is $m(\boldsymbol{s}) = \frac{1}{n}\sum_{j=1}^{n} s_j$, and the Boltzmann average magnetization is

$$\langle m \rangle_\mu = \sum_{\boldsymbol{s}} \mu(\boldsymbol{s})\, m(\boldsymbol{s}). \tag{10}$$

Eq. (10) involves a sum over all 2^n configurations, but given N samples $\{\boldsymbol{s}\} = \mathcal{S}$ from μ, the approximation $\langle m \rangle_\mu \approx N^{-1} \sum_{\boldsymbol{s} \in \mathcal{S}} m(\boldsymbol{s})$ can be accurate with high probability even if $N \ll 2^n$ [43]. While sampling from μ exactly may be infeasible, it is common to approximate $\langle m \rangle_\mu$ by the running average of $m(\boldsymbol{s})$ over MCMC trajectories (of one or several independent chains), and likewise for other average quantities. The quality of this approximation after a fixed number of MCMC iterations reflects the Markov chains' convergence rate [12].

We used this approach to estimate $\langle m \rangle_\mu$ at $T = 0.1$ as shown in Fig. 4. At this temperature $\langle m \rangle_\mu \approx 0.15$, and the Boltzmann probabilities of the ground (i.e., lowest-E), 1st, 2nd and 3rd excited configurations are approximately 43%, 26%, 19% and 12% respectively. This T is therefore sufficiently high that sampling from μ is not simply an optimization problem (where $\langle m \rangle_\mu$ depends overwhelmingly on the ground configuration), but sufficiently low that $\langle m \rangle_\mu$ is mostly determined by a few low-E configurations out of $2^n = 1024$. Efficiently estimating $\langle m \rangle_\mu$ using MCMC therefore involves finding these configurations and—crucially—jumping frequently between them in proportion to their Boltzmann probabilities. The magnetization $m(\boldsymbol{s})$ for illustrative trajectories is shown in Fig. 4a for the local and uniform proposal strategies, and for an experimental implementation of our quantum algorithm. While each Markov chain finds a low-E configurations quickly, our quantum

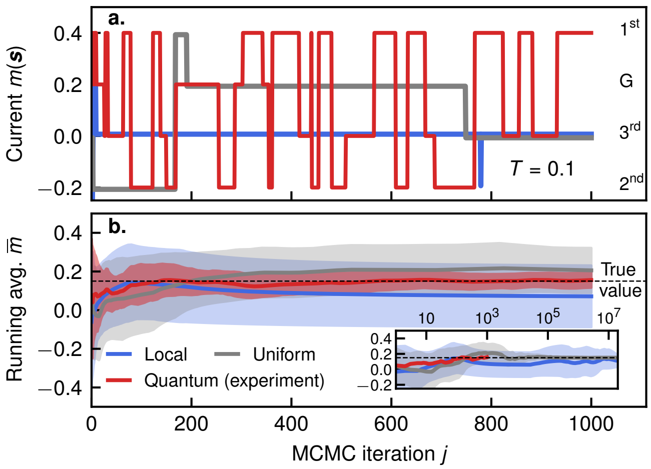


FIG. 4. **Magnetization estimate experiment. a.** The current magnetization $m(\boldsymbol{s}^{(j)})$ for individual Markov chains after j iterations. Each chain illustrates a different proposal strategy with uniformly random initialization. Arrows indicate the magnetization of the ground (G), 1st, 2nd and 3rd excited configurations. **b.** Convergence of the running average $\bar{m}^{(j)} = \frac{1}{j}\sum_{k=0}^{j} m(\boldsymbol{s}^{(k)})$ from MCMC trajectories to the true value of $\langle m \rangle_\mu$ for different proposal strategies. For each strategy, the lines and error bands show the mean and standard deviation, respectively, of $\bar{m}^{(j)}$ over 10 independent chains. The inset depicts the same chains over more iterations. We do not use a burn-in period or thinning (i.e., the running average starts at $k = 0$ and includes every iteration up to $k = j$), as these practices would introduce hyperparameters that complicate the interpretation. Both panels are for the same illustrative $n = 10$ instance at $T = 0.1$, and use the M-H acceptance probability (3).

algorithm jumps between these configurations noticeably faster than the others. The running average estimate for $\langle m \rangle_\mu$ from 10 independent Markov chains of each type is shown in Fig. 4b. Our quantum algorithm converges to the true value of $\langle m \rangle_\mu$, with no discernible bias, substantially faster than the two classical alternatives despite experimental imperfections.

Finally, we examined the mechanism underlying this observed quantum speedup. Recall that the local proposal strategy typically achieves small $|\Delta E| = |E(\boldsymbol{s}') - E(\boldsymbol{s})|$ by picking $\boldsymbol{s}'$ from the neighbors of $\boldsymbol{s}$, whereas the uniform proposal strategy typically picks $\boldsymbol{s}'$ far from $\boldsymbol{s}$ at the cost of a larger ΔE, as illustrated in Fig. 1c. Our quantum algorithm was motivated by the possibility of achieving the best of both: small $|\Delta E|$, and $\boldsymbol{s}'$ far from $\boldsymbol{s}$. This combination of features is illustrated in Fig. 1c and borne out in Fig. 5. The proposal probabilities $Q(\boldsymbol{s}'|\,\boldsymbol{s})$ arising in our algorithm for the same $n = 10$ model instance are shown in Figs. 5a-b for theory and experiment respectively. Both show the same effect with good qualitative agreement: starting from $\boldsymbol{s}$, our quantum algorithm mostly proposes jumps to configurations $\boldsymbol{s}'$ for which $|\Delta E|$ is small, even though $\boldsymbol{s}$ and $\boldsymbol{s}'$ may be far in Hamming distance. This effect is especially pronounced between the lowest-E configurations and also between highest-E ones.

To further examine this effect we asked the following: for a uniformly random configuration $\boldsymbol{s}$, what is the probability of proposing a $\boldsymbol{s} \to \boldsymbol{s}'$ jump for which $\boldsymbol{s}$ and $\boldsymbol{s}'$ are separated by a Hamming distance d, or by an energy difference $|\Delta E|$? The resulting distributions are shown in Figs. 5c-d respectively for the local and uniform proposal strategies, as well as for the theoretical and experimental realizations of our quantum algorithm. The Hamming distance distribution for local proposals is concentrated at $d = 1$ (by definition), whereas it is much more evenly spread for both uniform proposals and for our quantum algorithm, as shown in Fig. 5c. Conversely, the ΔE distribution of local proposals is more concentrated at small ΔE than that of uniform proposals. In theory, the corresponding distribution for our quantum algorithm is even more strongly concentrated at small ΔE, as shown in Fig. 5d. The ΔE distribution from the experimental realization of our algorithm, however, lies between those of the local and uniform proposal strategies, due to experimental imperfections.

OUTLOOK

Current quantum computers can sample from complicated probability distributions. We proposed and experimentally demonstrated a new quantum algorithm which leverages this ability in order to sample from the low-temperature Boltzmann distribution of classical Ising models, which is useful in many applications—not just complicated. Our algorithm uses relatively simple and shallow quantum circuits, thus enabling a quantum speedup on current hardware despite experimental imperfections. It works by alternating between quantum and classical steps on a shot-by-shot basis, unlike variational quantum algorithms, which typically run a quantum circuit many times in each step [44]. Rather, it uses a quantum computer to propose a random bit-string, which is accepted or rejected by a classical computer. The resulting Markov chain is guaranteed to converge to the desired Boltzmann distribution, even though it may not be possible to efficiently simulate classically. In this sense our algorithm is partially heuristic, like most classical MCMC algorithms: the eventual result is theoretically guaranteed, while fast convergence is established empirically.

Many state-of-the-art MCMC algorithms build upon simpler Markov chains in heuristically-motivated ways; for instance, by running several local M-H chains in parallel at different temperatures and occasionally swapping them [18, 19, 45]. Our quantum algorithm may provide a potent new ingredient for such composite algorithms in the near term. However, there remain ample oppor-

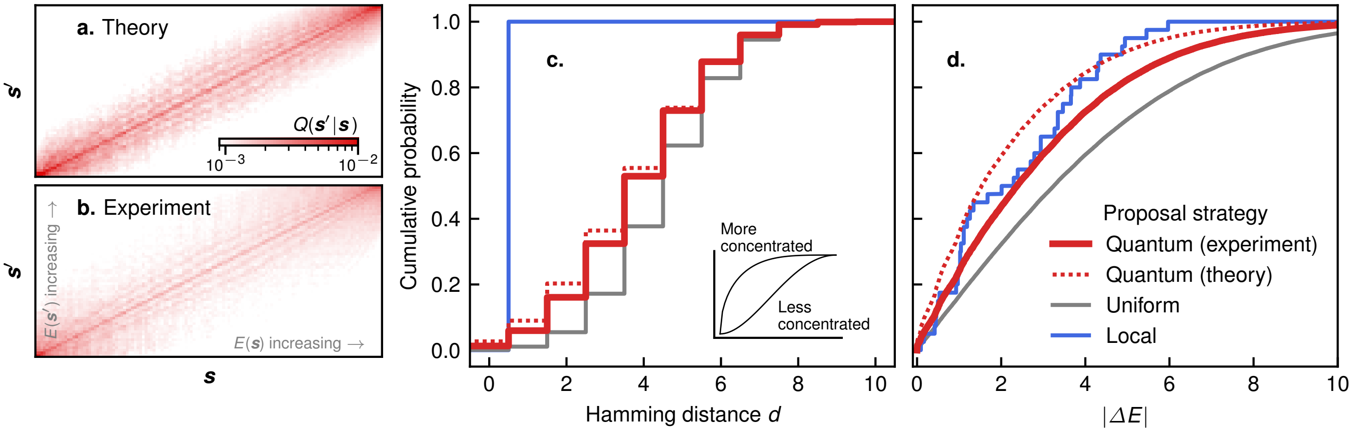


FIG. 5. **Quantum speedup mechanism.** **a.** The classically-simulated probabilities of $\boldsymbol{s} \to \boldsymbol{s'}$ proposals in our quantum algorithm, represented as a $2^n \times 2^n$ matrix whose columns are independent histograms. Both the initial and proposed configurations are sorted by increasing Ising energy E. **b.** The estimated proposal probabilities for our algorithm's experimental realization. We estimated each $Q(\boldsymbol{s'}|\,\boldsymbol{s})$ by the number of observed $\boldsymbol{s} \to \boldsymbol{s'}$ proposals normalized by the number of $\boldsymbol{s} \to$ [anything] proposals, using a total of 5.76×10^7 recorded quantum transitions. **c.** The probability distributions of Hamming distance between current ($\boldsymbol{s}$) and proposed ($\boldsymbol{s'}$) configurations, for a uniformly random current configuration. That of the experiment uses the estimated probabilities from panel b, while the rest were computed exactly. **d.** The analogous distributions for $|\Delta E| = |E(\boldsymbol{s'}) - E(\boldsymbol{s})|$ of proposed jumps. Each distribution is depicted in full detail through its cumulative distribution function, with no binning. All panels are for the same illustrative $n = 10$ model instance; none depend on T or on the choice of acceptance probability.

tunities for refinements and variations. For instance, a more targeted method of picking the parameters γ and t could further accelerate convergence [46]. Indeed, t did not depend on the problem size n in our implementation, although in some settings it should grow with n if the qubits are all to remain within each others' light cones. Moreover, different quantum processors with different connectivities, such as quantum annealers, may also be well-suited to implement our algorithm, perhaps without needing to discretize the Hamiltonian dynamics in Step 1.

Our algorithm is remarkably robust against imperfections. It achieves a speedup by proposing good jumps—but not every jump needs to be especially good for the algorithm to work well. (This is why picking γ and t at random works well: good values arise often enough.) For instance, if an error occurs in the quantum processor while a jump is being proposed, the proposal will be accepted/rejected as usual, and the Markov chain can still converge to the target distribution provided such errors do not break the $Q(\boldsymbol{s'}|\,\boldsymbol{s}) = Q(\boldsymbol{s}\,|\boldsymbol{s'})$ symmetry on average [11]. Rather than produce the wrong result, we found that such errors merely slow the convergence at low T by making our algorithm more classical. Indeed, in the limit of fully depolarizing noise, our algorithm reduces to MCMC with a uniform proposal. Our simulations suggest that the quantum speedup increases with the problem size n. However, we also expect the quantum noise to increase with n, in the absence of error correction, as the number of potential errors grows. The combined effect of these competing factors at larger n is currently unknown. It is interesting to note, however, that the cubic/quartic speedup we observed, should it persist at larger n, might give a quantum advantage on a modest fault-tolerant quantum computer despite the error correction overhead [34, 47].

Characterizing our algorithm at larger scales will require different methods than those employed here. For instance, a Markov chain's absolute spectral gap is a broad and unambiguous figure of merit, but it is not feasible to measure for large instances. This not an issue with our quantum algorithm in particular, but rather, with MCMC in general. Instead, a more fruitful approach may be to focus directly on how well our algorithm performs in various applications, such as in simulated annealing (for combinatorial optimization), for estimating thermal averages in many-body physics models, or for training and sampling from (classical) Boltzmann machines for machine learning applications.

ACKNOWLEDGEMENTS

We thank Easwar Magesan and Kristan Temme for useful discussions which helped shape the project, as well as Edward Chen, Nate Earnest-Noble, Andrew Eddins and Daniel Egger for crucial help with the experiments.

AUTHOR CONTRIBUTIONS

D.L. led the theory and the experiments. G.M., R.M., M.M. and P.W. contributed to the theory; in particular, D.L. and G.M. independently proposed a variant of this algorithm which uses quantum phase estimation, described in Section V-B of [11]. J.S.K., G.M., R.M., M.M. and S.S. contributed to the design of the experiments, and S.S. also contributed to their implementation. D.L. drafted the manuscript and supplemental material; all authors contributed to revising both.

[*] david.layden@ibm.com
[†] Current affiliation: NVIDIA

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).
[2] Y. Wu *et al.*, Strong quantum computational advantage using a superconducting quantum processor, Phys. Rev. Lett **127**, 180501 (2021).
[3] H.-S. Zhong *et al.*, Phase-programmable Gaussian boson sampling using stimulated squeezed light, Phys. Rev. Lett **127**, 180502 (2021).
[4] E. Ising, Beitrag zur Theorie des Ferromagnetismus, Z. Phys **31**, 253 (1925).
[5] K. Huang, *Statistical mechanics* (John Wiley & Sons, 2008).
[6] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, Cogn. Sci **9**, 147 (1985).
[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).
[8] A. Lucas, Ising formulations of many NP problems, Front. Phys **2** (2014).
[9] F. Barahona, On the computational complexity of Ising spin glass models, J. Phys. A **15**, 3241 (1982).
[10] J. Dongarra and F. Sullivan, Guest editors' introduction to the top 10 algorithms, Comput. Sci. Eng **2**, 22 (2000).
[11] Supplemental Material.
[12] D. Levin and Y. Peres, *Markov Chains and Mixing Times*, MBK (American Mathematical Society, 2017).
[13] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Comp. Phys. **21**, 1087 (1953).
[14] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika **57**, 97 (1970).
[15] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, An introduction to MCMC for machine learning, Mach. Learn **50**, 5 (2003).
[16] R. H. Swendsen and J.-S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, Phys. Rev. Lett **58**, 86 (1987).
[17] U. Wolff, Collective Monte Carlo updating for spin systems, Phys. Rev. Lett **62**, 361 (1989).
[18] J. Houdayer, A cluster Monte Carlo algorithm for 2-dimensional spin glasses, Eur. Phys. J. B **22**, 479 (2001).
[19] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, Efficient cluster algorithm for spin glasses in any space dimension, Phys. Rev. Lett **115**, 077201 (2015).
[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
[21] C. L. Baldwin and C. R. Laumann, Quantum algorithm for energy matching in hard optimization problems, Phys. Rev. B **97**, 224201 (2018).
[22] V. N. Smelyanskiy, K. Kechedzhi, S. Boixo, S. V. Isakov, H. Neven, and B. Altshuler, Nonergodic delocalized states for efficient population transfer within a narrow band of the energy landscape, Phys. Rev. X **10**, 011017 (2020).
[23] V. N. Smelyanskiy, K. Kechedzhi, S. Boixo, H. Neven, and B. Altshuler, Intermittency of dynamical phases in a quantum spin glass, arXiv:1907.01609 (2019).
[24] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).
[25] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete, Quantum Metropolis sampling, Nature **471**, 87 (2011).
[26] M.-H. Yung and A. Aspuru-Guzik, A quantum–quantum Metropolis algorithm, Proc. Natl. Acad. Sci **109**, 754 (2012).
[27] J. E. Moussa, Measurement-based quantum Metropolis algorithm, arXiv:1903.01451 (2019).
[28] P. Wocjan and K. Temme, Szegedy walk unitaries for quantum maps, arXiv preprint arXiv:2107.07365 (2021).
[29] M. Szegedy, Quantum speed-up of Markov chain based algorithms, in *45th Annual IEEE Symposium on Foundations of Computer Science* (2004) pp. 32–41.
[30] P. C. Richter, Quantum speedup of classical mixing processes, Phys. Rev. A **76**, 042306 (2007).
[31] R. D. Somma, S. Boixo, H. Barnum, and E. Knill, Quantum simulations of classical annealing processes, Phys. Rev. Lett. **101**, 130504 (2008).
[32] P. Wocjan and A. Abeyesinghe, Speedup via quantum sampling, Phys. Rev. A **78**, 042336 (2008).
[33] A. W. Harrow and A. Y. Wei, Adaptive quantum simulated annealing for Bayesian inference and estimating partition functions, in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (SIAM, 2020) pp. 193–212.
[34] J. Lemieux, B. Heim, D. Poulin, K. Svore, and M. Troyer, Efficient quantum walk circuits for Metropolis-Hastings algorithm, Quantum **4**, 287 (2020).
[35] S. Arunachalam, V. Havlicek, G. Nannicini, K. Temme, and P. Wocjan, Simpler (classical) and faster (quantum) algorithms for Gibbs partition functions, in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2021) pp. 112–122.
[36] D. S. Wild, D. Sels, H. Pichler, C. Zanoci, and M. D. Lukin, Quantum sampling algorithms for near-term devices, Phys. Rev. Lett. **127**, 100504 (2021).
[37] D. S. Wild, D. Sels, H. Pichler, C. Zanoci, and M. D. Lukin, Quantum sampling algorithms, phase transitions, and computational complexity, Phys. Rev. A **104**, 032602 (2021).
[38] D. Sherrington and S. Kirkpatrick, Solvable model of a spin-glass, Phys. Rev. Lett **35**, 1792 (1975).
[39] M. Anis Sajid *et al.*, Qiskit: An open-source framework for quantum computing (2021).
[40] M. Suzuki, Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics, J. Math. Phys **26**, 601 (1985).
[41] J. J. Wallman and J. Emerson, Noise tailoring for scalable quantum computation via randomized compiling, Phys. Rev. A **94**, 052325 (2016).

[42] N. Earnest, C. Tornow, and D. J. Egger, Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware, Phys. Rev. Research **3**, 043088 (2021).

[43] V. Ambegaokar and M. Troyer, Estimating errors reliably in Monte Carlo simulations of the Ehrenfest model, Am. J. Phys **78**, 150 (2010).

[44] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, Variational quantum algorithms, Nat. Rev. Phys **3**, 625 (2021).

[45] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo simulation of spin-glasses, Phys. Rev. Lett **57**, 2607 (1986).

[46] G. Mazzola, Sampling, rates, and reaction currents through reverse stochastic quantization on quantum computers, Phys. Rev. A **104**, 022431 (2021).

[47] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, Focus beyond quadratic speedups for error-corrected quantum advantage, PRX Quantum **2**, 010103 (2021).

Supplementary information for: “Quantum-enhanced Markov chain Monte Carlo”

David Layden,[1,*] Guglielmo Mazzola,[2] Ryan V. Mishmash,[1] Mario Motta,[1] Pawel Wocjan,[3] Jin-Sung Kim,[1,†] and Sarah Sheldon[1]

[1] *IBM Quantum, Almaden Research Center, San Jose, California 95120, USA*
[2] *IBM Quantum, IBM Research – Zurich, 8803 Rüschlikon, Switzerland*
[3] *IBM Quantum, T. J. Watson Research Center, Yorktown Heights, NY 10598, USA*

CONTENTS

I. Notation 2

II. Algorithm details 2
A. Irreducibility and aperiodicity 3

III. Numerical details 3
A. Implementation 3
B. Random instances and connectivity 4
C. Supplemental data 5
1. 1D model instances 5
2. Different n and T 6
3. Gibbs sampler acceptance probability 8
4. Lazy chains 11
D. Cluster Algorithms 13
1. Swendsen-Wang and Wolff clusters 13
2. Houdayer clusters 14

IV. Experimental details 15
A. Quantum circuits 15
1. Trotter circuits and gates 15
2. Twirling 17
3. Overall structure of experiments 19
B. Data analysis 20
1. MCMC trajectories 20
2. Symmetry testing 21
3. Bootstrapping 22
C. Supplemental data 23
1. $n = 10$ instance 24
2. $n = 9$ instance 27
3. $n = 8$ instance 31

V. Quantum proposal strategies: motivation and alternatives 35
A. Perturbative regime 35
B. Quantum phase estimation 36
C. Reverse annealing 37

References 39

[*] david.layden@ibm.com
[†] Current affiliation: NVIDIA

I. NOTATION

We will use $\boldsymbol{P}$ to denote a $2^n \times 2^n$ transition matrix formed from transition probabilities $\{P(\boldsymbol{s'} \,|\, \boldsymbol{s})\}$ in lexicographic order, and likewise $\boldsymbol{Q}$ for a $2^n \times 2^n$ matrix of proposal probabilities $\{Q(\boldsymbol{s'} \,|\, \boldsymbol{s})\}$. We will also use $\vec{\mu} \in \mathbb{R}^{2^n}$ for the vector of probabilities $\{\mu(\boldsymbol{s})\}$ with the same ordering. To simplify the notation, we will index both with integers in $[0, 2^n - 1]$ whose binary representation encodes a spin configuration, e.g.:

Integer	Spin configuration
$j = 0$	$\boldsymbol{s} = (1, \ldots, 1, 1)$
$j = 1$	$\boldsymbol{s} = (1, \ldots, 1, -1)$
$j = 2$	$\boldsymbol{s} = (1, \ldots, -1, 1)$
$\vdots$	$\vdots$
$j = 2^n - 1$	$\boldsymbol{s} = (-1, \ldots, -1, -1)$

We will refer to a spin configuration by its vector ($\boldsymbol{s} \in \{1, -1\}^n$) or integer ($j \in [0, 2^n - 1]$) representation interchangeably. Moreover, to be consistent with the convention used in quantum mechanics, we will treat $\vec{\mu}$ as a column vector, and $\boldsymbol{P}$ and $\boldsymbol{Q}$ as left-stochastic matrices, meaning their columns sum to 1.

II. ALGORITHM DETAILS

Our full algorithm is shown in Algorithm S1. This version uses the Metropolis-Hastings (M-H) acceptance probability from Eq. (3) of the main text on line 9, although any alternative satisfying detailed balance would also work. In Section III C 3, for instance, we consider the Gibbs sampler acceptance probability, but find little difference in performance. Gauging MCMC convergence can be difficult in practice, and is usually done heuristically [1]. This is a property of MCMC in general, regardless of whether moves are proposed by classical or quantum means. For the sake of clarity then, Algorithm S1 assumes that the user already has a convergence diagnostic in mind that is well-suited for the application at hand. We will occasionally write the Hamiltonian H from Eq. (5) of the main text as $H(\gamma)$ to emphasize its dependence on the parameter $\gamma \in [0, 1]$.

Algorithm S1: Quantum-enhanced Markov chain Monte Carlo

```
1  $\boldsymbol{s}$ = initial spin configuration (often chosen uniformly at random)
2  while not converged:
3      Propose jump (quantum step 1)
4      $\gamma$ = random.uniform(0.25, 0.6)
5      $t$ = random.uniform(2, 20)
6      $|\psi\rangle = \exp[-iH(\gamma)\, t]|\boldsymbol{s}\rangle$ on quantum device
7      $\boldsymbol{s'}$ = result of measuring $|\psi\rangle$ in computational basis
8      Accept/reject jump (classical step 2)
9      $A = \min(1, e^{[E(\boldsymbol{s}) - E(\boldsymbol{s'})]/T})$
10     if $A \geq$ random.uniform(0, 1):
11         $\boldsymbol{s} = \boldsymbol{s'}$
```

As discussed in the main text, we have not sought to optimize the free parameters γ and t in our quantum proposal mechanism, as the optimal values seem to have a complicated dependence on the model instance. Rather, we sampled them randomly in each MCMC iteration according to $\gamma \sim$ uniform([0.25, 0.6]) and $t \sim$ uniform([2, 20]). We settled on these *ad hoc* distributions through trial and error on a small number of fully-connected random model instances, and found them to be robust across a wide range of instances for various n and model connectivities. For a given instance, certain values of (γ, t) might lead to particularly good MCMC moves (e.g., between distant local minima) while others do not. Picking (γ, t) at random ensures that these good values (or nearby ones) arise regularly, even if we don't know what they are *a priori*. Accordingly, some fraction of the moves will be particularly good, which proves sufficient for a quantum enhancement in the absolute spectral gap δ. This randomized strategy works reasonably well, although more sophisticated strategies are a promising area of future research.

A. Irreducibility and aperiodicity

We begin with some basic definitions and results concerning Markov chains. A Markov chain with transition matrix $\boldsymbol{P}$ is said to be irreducible if for all $j, k \in [0, 2^n - 1]$ there exists an integer $m > 0$ (possibly depending on j and k) such that $(\boldsymbol{P}^m)_{jk} > 0$. Let

$$\mathcal{M}(j) = \{m \geq 1 \,|\, (\boldsymbol{P}^m)_{jj} > 0\}, \tag{S1}$$

then the period of configuration j is defined to be the greatest common divisor of $\mathcal{M}(j)$. A Markov chain is called aperiodic if all configurations j have period 1, otherwise it is called periodic [2] (Section 1.3). A distribution $\vec{\mu}$ is said to be stationary for a Markov chain if $\boldsymbol{P}\vec{\mu} = \vec{\mu}$. If $\vec{\mu}$ satisfies the detailed balance condition for a Markov chain, Eq. (2) of the main text, then $\vec{\mu}$ is a stationary distribution of the chain [2] (Proposition 1.19). An irreducible Markov chain has a unique stationary distribution $\vec{\mu}$ [2] (Corollary 1.17). Finally, if a Markov chain is irreducible and aperiodic with stationary distribution $\vec{\mu}$ then any initial distribution of the chain will converge to $\vec{\mu}$ [2] (Theorem 4.9). Putting this all together, an irreducible and aperiodic Markov chain that satisfies detailed balance with respect to $\vec{\mu}$ will converge to $\vec{\mu}$, as stated in the main text. This a powerful result, as it guarantees convergence to $\vec{\mu}$ for all initial states, under mild conditions, even if $\boldsymbol{P}$ is too large to diagonalize (or store in memory).

Detailed balance is satisfied by design in our algorithm through the use of an appropriate acceptance probability, e.g., Eq. (3) of the main text. (In that equation we assumed for simplicity that $Q(\boldsymbol{s'}|\boldsymbol{s}) \neq 0$ for an $\boldsymbol{s} \to \boldsymbol{s'}$ proposal. If $Q(\boldsymbol{s'}|\boldsymbol{s}) = 0$ then the value of $A(\boldsymbol{s'}|\boldsymbol{s})$ is irrelevant since there will never be an opportunity to accept or reject a $\boldsymbol{s} \to \boldsymbol{s'}$ proposal to begin with.) What remains to be shown is that our algorithm produces an irreducible and aperiodic Markov chain.

Suppose that $Q(\boldsymbol{s'}|\boldsymbol{s}) > 0$ for all configurations $\boldsymbol{s}$ and $\boldsymbol{s'}$. Since $\mu(\boldsymbol{s}) > 0$ for all $\boldsymbol{s}$ at nonzero temperature, it follows from Eq. (3) of the main text that $A(\boldsymbol{s'}|\boldsymbol{s}) > 0$ and therefore $P(\boldsymbol{s'}|\boldsymbol{s}) > 0$ for all $\boldsymbol{s}$ and $\boldsymbol{s'}$. The corresponding Markov chain is therefore irreducible (with $m = 1$) by definition. Similarly, since $Q(\boldsymbol{s}|\boldsymbol{s}) > 0$ by assumption and $A(\boldsymbol{s}|\boldsymbol{s}) = 1$ for all $\boldsymbol{s}$, $P(\boldsymbol{s}|\boldsymbol{s}) > 0$ so every state $\boldsymbol{s}$ has period 1 and the chain is aperiodic. In summary, a sufficient condition for the chain to be irreducible and aperiodic is for all the proposal probabilities $Q(\boldsymbol{s'}|\boldsymbol{s})$ to be nonzero. Formally, this could be guaranteed by proposing moves as follows: for some small $\epsilon > 0$, propose the next configuration $\boldsymbol{s'}$ uniformly at random (i.e., use the uniform proposal) with probability ϵ, otherwise propose $\boldsymbol{s'}$ using a quantum computer as in our algorithm. In practice, however, we found no need for this formality. Absent some special symmetry, no $\boldsymbol{s} \to \boldsymbol{s'}$ quantum transition should be completely forbidden. Considering also experimental noise, $Q(\boldsymbol{s'}|\boldsymbol{s}) > 0$ for all $\boldsymbol{s}$ and $\boldsymbol{s'}$ is all-but formally assured without having to incorporate the uniform proposal as described above. The resulting Markov chain therefore converges to $\vec{\mu}$ on a combination of physical and mathematical grounds.

III. NUMERICAL DETAILS

This section describes the numerics underlying Figs. 2, 3 and 5 of the main text (not including the experimental data in Figs. 3 and 5, which is treated in Section IV), and also presents a range of complementary numerical results.

A. Implementation

We analyzed the spectral gap δ for the local, uniform, quantum and mismatched quantum proposals as follows. For a given model instance and temperature T, we explicitly computed every proposal probability $Q(\boldsymbol{s'}|\boldsymbol{s})$ and in turn each transition probability $P(\boldsymbol{s'}|\boldsymbol{s})$ (i.e., we did not approximate either through sampling). We then formed $\{P(\boldsymbol{s'}|\boldsymbol{s})\}$ into a $2^n \times 2^n$ transition matrix $\boldsymbol{P}$, which we numerically diagonalized to find δ. We repeated this procedure for each n, T, and model instance to form Fig. 2 of the main text. We applied the same steps for a single model instance in Fig. 3 of the main text. In Fig. 5 of the main text we only computed $Q(\boldsymbol{s'}|\boldsymbol{s})$, which does not depend on T. For the mismatched quantum proposal, we picked the "mismatched" parameters $\tilde{J}_{jk}$ and $\tilde{h}_j$ from the same (standard normal) distribution as the actual parameters J_{jk} and h_k which define the model instance. For each model instance we picked new mismatched parameters. This approach gives meaningful results over an ensemble of model instances, but not for individual instances, since there could be significant variation in the "mismatched" δ depending on which $\tilde{J}_{jk}$ and $\tilde{h}_j$ are drawn. We therefore did not consider the mismatched quantum proposal in Figs. 3 and 5, nor in similar analyses below.

For our quantum algorithm, we simulated noiseless dynamics generated by H. Viewed as a quantum channel $\mathcal{C}$ acting on an initial state $\rho = |\boldsymbol{s}\rangle\langle\boldsymbol{s}|$, the quantum proposal mechanism described in Algorithm S1 is

CPTP map.

$$\mathcal{C}(\rho) = \frac{1}{(\gamma_{\max}-\gamma_{\min})(t_{\max}-t_{\min})} \int_{\gamma_{\min}}^{\gamma_{\max}} d\gamma \int_{t_{\min}}^{t_{\max}} dt \;\; U(\gamma,t)\,\rho\, U(\gamma,t)^\dagger, \tag{S2}$$

where $U(\gamma,t) = e^{-iH(\gamma)t}$, $[\gamma_{\min},\gamma_{\max}] = [0.25, 0.6]$, and $[t_{\min}, t_{\max}] = [2, 20]$. In our numerics, however, we implemented a slightly modified version of Algorithm S1 for simplicity, corresponding to a different channel $\mathcal{C}' \approx \mathcal{C}$, where γ is drawn from a discrete, rather than continuous, uniform distribution. The issue is that the integral over t in Eq. (S2) is easy to evaluate analytically, but we had to resort to numerical integration for the one over γ, which we approximated by a midpoint Riemann sum with 20 subintervals of equal size. (Increasing the number of subintervals made little difference to the results, suggesting a good approximation to Eq. (S2).) This amounts to sampling

$$\gamma \sim \text{uniform}\left(\left\{0.25 + \frac{\Delta\gamma}{2},\ 0.25 + \frac{3\Delta\gamma}{2},\ 0.25 + \frac{5\Delta\gamma}{2},\ \ldots,\ 0.6 - \frac{\Delta\gamma}{2}\right\}\right) \tag{S3}$$

for each MCMC step (Algorithm S1 line 4), with a subinterval width of $\Delta\gamma = 0.0175$, rather than $\gamma \sim \text{uniform}([0.25, 0.6])$. Algorithm S1 uses the latter distribution for conceptual simplicity, but in practice the choice seems to make little difference.

B. Random instances and connectivity

To create Fig. 2 of the main text and similar figures below, we generated random model instances by drawing coefficients J_{jk} and h_j that are independent and identically distributed (IID) from standard normal distributions (i.e., with zero mean and unit variance). Including non-zero fields $\{h_j\}$ of comparable size to the couplings serves to complicated the sampling problem: at low T, the spins seek to align with the local fields, but also to satisfy all the couplings. It is not typically possible to do both, and neither objective dominates. In effect, an n-spin system with non-zero fields $\{h_j\}$ is equivalent to an $(n+1)$-spin system without fields, where the original spins $j \in \{1, \ldots, n\}$ are instead also coupled with strength $J_{n+1,j} = h_j$ to an additional spin that is held fixed in the $+1$ state (sometimes called a "ghost spin"). For fully-connected models this makes little difference: a fully-connected n-spin model with fields amounts to a fully-connected $(n+1)$-spin model without fields. The difference is more pronounced when the connectivity is sparser, like in the instances we implemented experimentally. For example, an n-spin model with 1D nearest-neighbor connectivity and non-zero fields amounts to a field-less $(n+1)$-spin model which does not form a 1D lattice since each spin couples to its neighbors, as well as to a common spin.

Drawing J_{jk} and h_j from independent standard normal distributions gives an ensemble of model instances similar to the well-known Sherrington-Kirkpatrick (S-K) model [3], but with two main differences: (i) the original S-K model does not include the random local fields discussed above, and (ii) it draws $\{J_{jk}\}$ from normal distributions with standard deviation $1/\sqrt{n}$ rather than 1. (Or equivalently, it draws $\{J_{jk}\}$ from standard normal distributions but includes a $1/\sqrt{n}$ prefactor in the energy $E(\boldsymbol{s})$ that is not present in Eq. (1) of the main text.) Our variation of the S-K model is also common, see, e.g., Ref. [4]. We have chosen to use a distribution that does not depend on n for two main reasons. First, the S-K $1/\sqrt{n}$ scaling is used frequently in the physics literature because it ensures that the $n \to \infty$ thermodynamic limit is well-behaved. However, coefficients arising in other applications (e.g., in Boltzmann machines) need not exhibit such finely-tuned scaling. The second reason has to do with connectivity. Interpreting (J_{jk}) as the adjacency matrix of a graph whose vertices are spins and whose edges are couplings, picking all $\{J_{jk}\}_{j>k}$

at random—without fixing any to be zero—gives a fully-connected graph (almost surely). This is the case we focused on in our numerics. Other graphs of lower degree (where some of the J_{jk} are set to zero) are also of interest, however, such as the ones used in our experiments, where the spins form a 1D chain with nearest-neighbor connectivity. It is not always clear how the S-K $1/\sqrt{n}$ scaling should be generalized for models with different connectivity. We therefore chose to avoid such scaling altogether. However, it would be trivial to introduce this $1/\sqrt{n}$ scaling into our results by simply rescaling the temperature as $T \to T/\sqrt{n}$.

Note that the $1/\sqrt{n}$ scaling of the original S-K model arises naturally in the quantum part of our algorithm through the $\alpha = \|H_{\text{mix}}\|_{\text{F}}/\|H_{\text{prob}}\|_{\text{F}}$ scaling factor in Eq. (5) of the main text. To see how, suppose there are $N-n$ couplings $\{J_{jk}\}$ that are not fixed to 0, while the rest are IID as $J_{jk}, h_j \sim \mathcal{N}(0,\sigma^2)$ for some standard deviation σ. Let $\vec{x} \in \mathbb{R}^N$ be the vector formed by stacking the fields $\{h_j\}$ and the non-zero couplings, which follows a multivariate normal distribution $p(\vec{x}) = (2\pi\sigma^2)^{-N/2}\exp(-\|\vec{x}\|^2/2\sigma^2)$. Integrating this probability density over $\vec{x} \in \mathbb{R}^N$ in N-dimensional spherical coordinates gives

$$1 = \int d^N x\, p(\vec{x}) = (2\pi\sigma^2)^{-N/2} \int_0^\infty dr\, e^{-r^2/2\sigma^2} r^{N-1} \int d\Omega = \frac{\Gamma(N/2)}{2\pi^{N/2}} \int d\Omega, \tag{S4}$$

where Γ denotes the gamma function, $r = \|\vec{x}\|$, and the differential solid angle $d\Omega$ is integrated over the full surface of an N-dimensional sphere. The average scaling factor can then be found by integrating α and solving for $\int d\Omega$ in the previous equation:

$$\langle\alpha\rangle_{J,h} = \sqrt{n} \int d^N x\, \|\vec{x}\|^{-1} p(\vec{x}) \; = \frac{\sqrt{n}}{(2\pi\sigma^2)^{N/2}} \int_0^\infty dr e^{-r^2/2\sigma^2} r^{N-2} \int d\Omega = \frac{\sqrt{n}}{\sigma\sqrt{2}}\, \Gamma\left(\frac{N-1}{2}\right) \Big/ \Gamma\left(\frac{N}{2}\right). \tag{S5}$$

For a fully-connected model (depicted in Fig. 1a of the main text) $N = n + \binom{n}{2}$, so asymptotically

$$\langle\alpha\rangle_{J,h} = \frac{\sqrt{2}}{\sigma\sqrt{n}} + O\left(\sigma^{-1} n^{-3/2}\right). \tag{S6}$$

Setting $\sigma = 1$ as in our numerics, H_{prob}, and in turn $E(\boldsymbol{s})$, gets scaled by a factor of $\sqrt{2/n}$ on average in Eq. (5). Had we used the S-K convention of $\sigma = 1/\sqrt{n}$, H_{prob} would only get scaled be $\sqrt{2}$, producing the same result. In other words, for the quantum part of our algorithm it doesn't matter whether we use the conventional $1/\sqrt{n}$ S-K scaling since it arises naturally in Eq. (5) regardless.

For models with different connectivity the story is different. A 1D chain of n spins with nearest-neighbor couplings (depicted in Fig. 1b of the main text), as in our experiments, has $N = 2n-1$ so

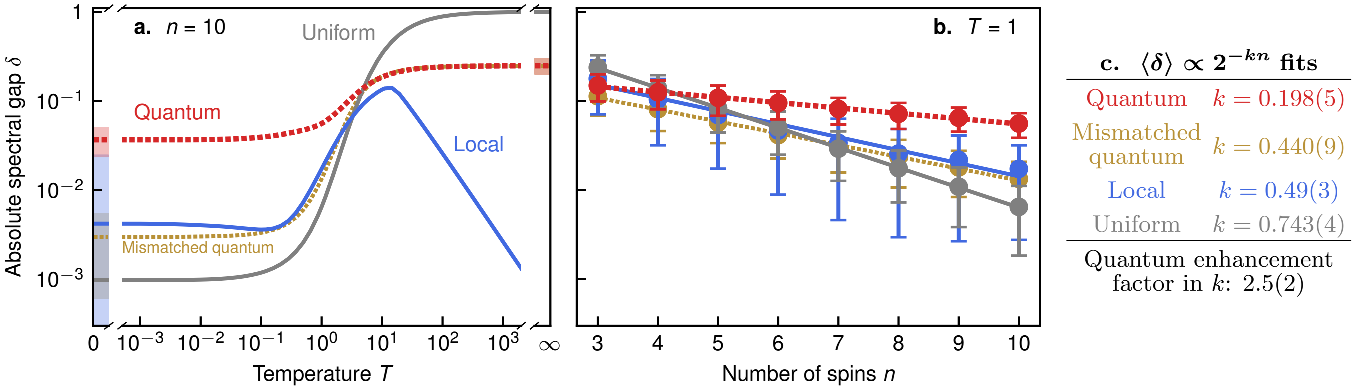
$$\langle\alpha\rangle_{J,h} = \frac{1}{\sigma\sqrt{2}} + O\left(\sigma^{-1} n^{-1}\right), \tag{S7}$$

meaning the scaling factor in Eq. (5) tends towards $1/\sqrt{2}$ on average for large n when $\sigma = 1$. We give numerical results for such 1D model instances in the next section.

C. Supplemental data

1. *1D model instances*

In the main text we describe our analysis of δ over many random fully-connected model instances, summarized in Fig. 2. We also performed a similar analysis for 1D instances (with open boundaries), which we present here. We picked $J_{j+1,j}$ and h_j IID from standard normal distributions, $\mathcal{N}(0,1)$, and fixed all other J_{jk} to zero. Fig. S1 shows the analogous results to Fig. 2 of the main text but for 500 random instances per n with 1D nearest-neighbor connectivity. The mismatched quantum proposal also uses $\tilde{J}_{jk}$ with 1D connectivity; that is, $\tilde{J}_{j+1,j}, \tilde{h}_j \sim \mathcal{N}(0,1)$ all IID, and all other $\tilde{J}_{jk} = 0$. For clarity, all other settings are identical to those used in Fig. 2. The average δ (which we denote $\langle\delta\rangle$) vs. T curve for our quantum algorithm in Fig. S1a remains very similar to that in Fig. 2a for fully-connected instances. The main difference is in $\langle\delta\rangle$ for the local proposal, which is much larger on 1D instances than on fully-connected ones. This is expected: sparser connectivity reduces the potential for frustration, thus increasing the fraction of easy instances. The scaling advantage of $\langle\delta\rangle$ versus n at $T = 1$ from the main text persists in Fig. S1b, though it is less pronounced due to enhanced performance of the local proposal. Note, finally, that care must be taken when comparing such plots across different model connectivities, since the typical scale of $E(\boldsymbol{s})$ depends on the number of nonzero J_{jk} coefficients. This effectively rescales the temperature in a way that depends on n, thus introducing some ambiguity when comparing results for different connectivities at a fixed T.



c. $\langle\delta\rangle \propto 2^{-kn}$ **fits**

Quantum	$k = 0.198(5)$
Mismatched quantum	$k = 0.440(9)$
Local	$k = 0.49(3)$
Uniform	$k = 0.743(4)$

Quantum enhancement factor in k: 2.5(2)

FIG. S1. **Average-case convergence rate simulations for 1D model instances.** Analogous data to that shown in Fig. 2 of the main text, but for 1D model instances rather than fully-connected ones. All strategies were simulated classically. Lines/markers show the average δ over 500 random 1D Ising model instances for each n; error bands/bars show the standard deviation in δ over these instances. Dotted lines are for visibility. **a.** The slow-down of each strategy at low T. The local proposal strategy performs better on average here than in Fig. 2a due to the larger fraction of easy instances. The change in connectivity has only a minor effect on δ for the other strategies. **b.** Problem size dependence, with least squares exponential fits to the average δ, weighted by the standard error of the mean. **c.** The resulting fit parameters and the average quantum enhancement exponent, which is the ratio of k for the quantum algorithm and the smallest k among classical proposal strategy (the local strategy, here). Uncertainties are from the fit covariance matrices.

2. *Different n and T*

We now return to fully-connected model instances, and further analyze the numerical data underlying Fig. 2 of the main text. Two parameters, n and T, control the average problem difficulty in our numerics. Fig. 2 in the main text shows two representative slices through n-T parameter space. Figs. S2 and S3 show the same quantities for other values of n and T. Each panel of Fig. S2 is qualitatively similar to Fig. 2a of the main text, but they show a growing separation in δ with increasing n at low T between our quantum algorithm and the purely classical Markov chains. Similarly, the panels of Fig. S3 show the onset of an apparent scaling advantage of δ with n, illustrated in Fig. 2b of the main text, which gets increasingly pronounced as T decreases. The exponential fit parameters (as in Fig. 2c of the main text) for Fig. S3 are given in Table S1.

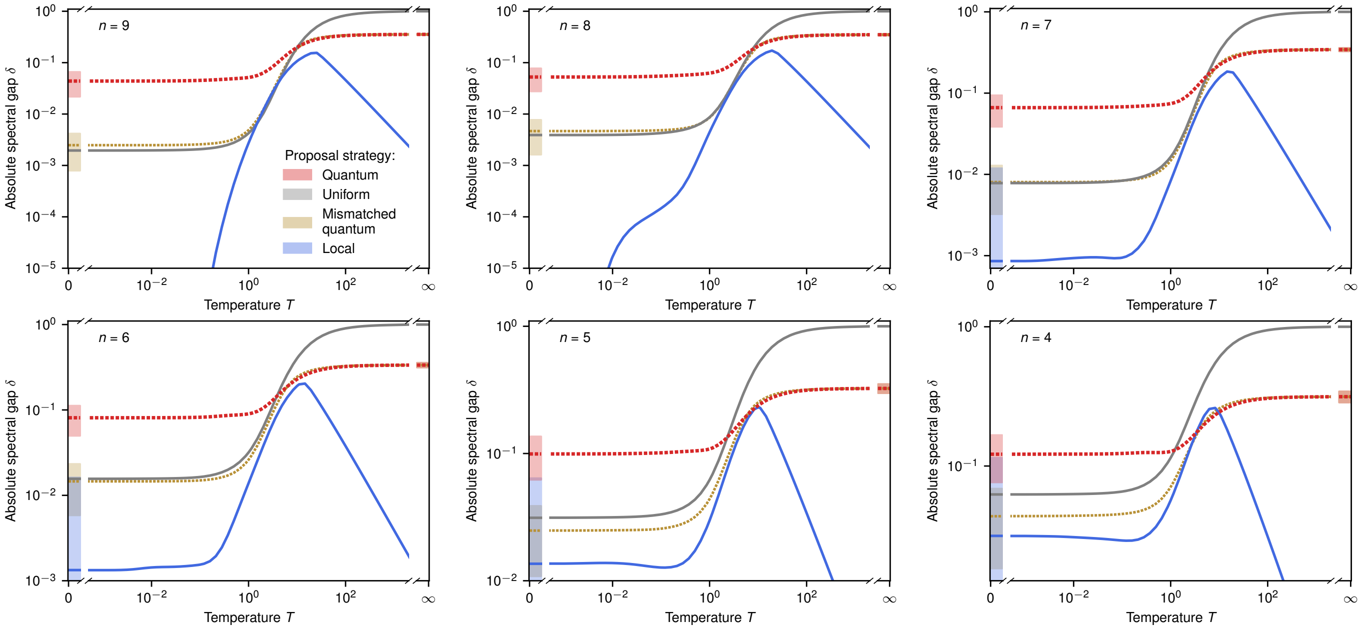


FIG. S2. **Average-case M-H convergence rate simulations for different n.** Analogous plots to Fig. 2a from the main text, but for different values of n. All strategies were simulated classically. Lines show the average δ over 500 random fully-connected Ising model instances for each n; error bands show the standard deviation in δ over these instances. Dotted lines are for visibility. The same model instances were used to make Fig. 2 of the main text.

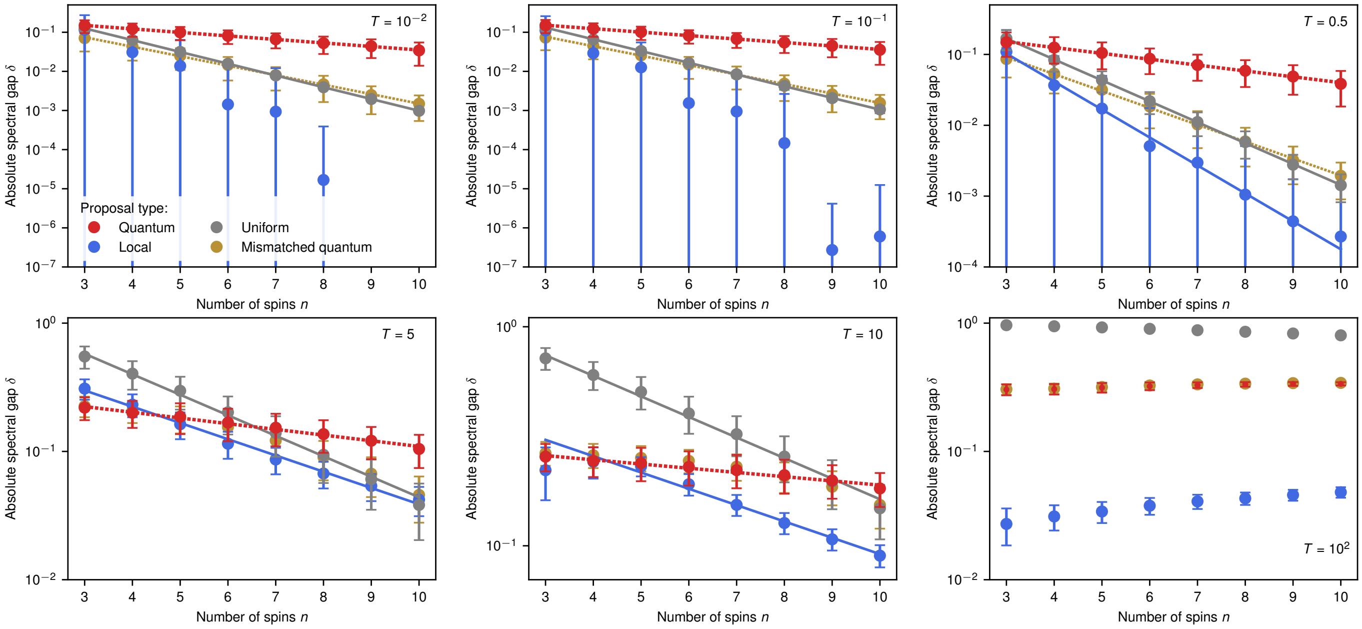


FIG. S3. **Average-case M-H convergence rate simulations for different T.** Analogous plots to Fig. 2b from the main text, but for different values of T. All strategies were simulated classically. Markers show the average δ over 500 random fully-connected Ising model instances for each n; error bars show the standard deviation in δ over these instances. Lines show least squares exponential fits $\langle\delta\rangle \propto 2^{-kn}$, weighted by the standard error of the mean, wherever such fits are reasonably good. The resulting values of k are given in Table S1. Dotted lines are for visibility. The same model instances were used to make Fig. 2 of the main text.

Proposal type	$T = 10^{-2}$	$T = 10^{-1}$	$T = 0.5$	$T = 5$	$T = 10$	$T = 10^{2}$
Quantum	$k = 0.299(3)$	$k = 0.293(2)$	$k = 0.275(4)$	$k = 0.147(6)$	$k = 0.064(4)$	-
Mismatched quantum	$k = 0.806(9)$	$k = 0.803(8)$	$k = 0.792(9)$	-	-	-
Local	-	-	$k = 1.31(4)$	$\boldsymbol{k = 0.42(1)}$	$\boldsymbol{k = 0.25(2)}$	-
Uniform	$\boldsymbol{k = 0.9999(2)}$	$\boldsymbol{k = 0.995(3)}$	$\boldsymbol{k = 0.981(4)}$	$k = 0.53(2)$	$k = 0.31(1)$	-
Quantum enhancement	3.35(3)	3.39(3)	3.57(6)	2.8(1)	3.9(4)	-

TABLE S1. **M-H $\langle\delta\rangle$ vs. n fits on fully-connected instances.** The exponential fit parameters k from $\langle\delta\rangle \propto 2^{-kn}$ in Fig. S3 for different proposal strategies, wherever such fits are reasonably good. The resulting average quantum enhancement exponent for each T, which is the ratio of k for the quantum algorithm and the smallest k among classical proposal strategies (shown in bold), is given in the bottom row. Uncertainties are from the fit covariance matrices.

3. *Gibbs sampler acceptance probability*

The results presented above and in the main text use the Metropolis-Hastings (M-H) acceptance probability from Eq. (3), which we will denote here as A_{MH}. As stated in the main text, however, there is a continuum of possible acceptance probabilities that satisfy detailed balance [5]. Yet, to the best of our knowledge, the only two that seem to be used frequently in practice are A_{MH} and

$$A_{\text{G}}(\boldsymbol{s}'|\boldsymbol{s}) = \left[1 + \left(\frac{\mu(\boldsymbol{s}')}{\mu(\boldsymbol{s})}\frac{Q(\boldsymbol{s}\,|\boldsymbol{s}')}{Q(\boldsymbol{s}'|\,\boldsymbol{s})}\right)^{-1}\right]^{-1}, \tag{S8}$$

which is variously named after Gibbs (hence the G subscript), Glauber, Boltzmann and Barker. The Markov chain resulting from A_{G} with a local proposal is called a Gibbs sampler; we will therefore refer to Eq. (S8) as the Gibbs sampler acceptance probability. (The Gibbs sampler algorithm is sometimes expressed in a way that combines the proposal and accept/reject steps into a single step [6, 7]. In the context of the Ising model, however, this "rejection-free" version is equivalent to proposing local jumps and accepting them with probability A_{G}; see [8] §4.4.) Neither A_{MH} nor A_{G} is strictly better than the other. A_{MH} is always larger than A_{G}, so it tends to produce slightly faster convergence at low T, where frequent rejections are the main bottleneck. (The difference is typically small, however, since the two approach each other as $T \to 0$.) We focused on M-H because we are primarily interested in the hard, low-T regime. However, whereas $A_{\text{MH}} \to 1$ for all moves as $T \to \infty$, $A_{\text{G}} \to 1/2$ in the same limit. This can actually accelerate convergence compared to M-H, notably for local proposals at high T. The reason is that the M-H transition matrix for a local proposal at high T, $\boldsymbol{P}_{\text{MH}}$, is nearly periodic and has an eigenvalue near -1. Using the Gibbs sampler remedies this issue (cf. Eq. (S1)) and prevents the spectral gap from vanishing as $T \to \infty$.

While we used A_{MH} in Algorithm S1, we could just as well have used A_{G} (or any other acceptance probability with similar dependence on μ and Q satisfying detailed balance) [5]. We found this choice to have little impact on δ in the low-T regime of interest, with A_{G} typically producing marginally slower convergence than A_{MH}, as shown in Figs. S4 and S5, and Table S2. Crucially, the arguments in the main text and in Section II A about irreducibility, aperiodicity, and the computational efficiency of evaluating A_{MH} hold straightforwardly for A_{G} too.

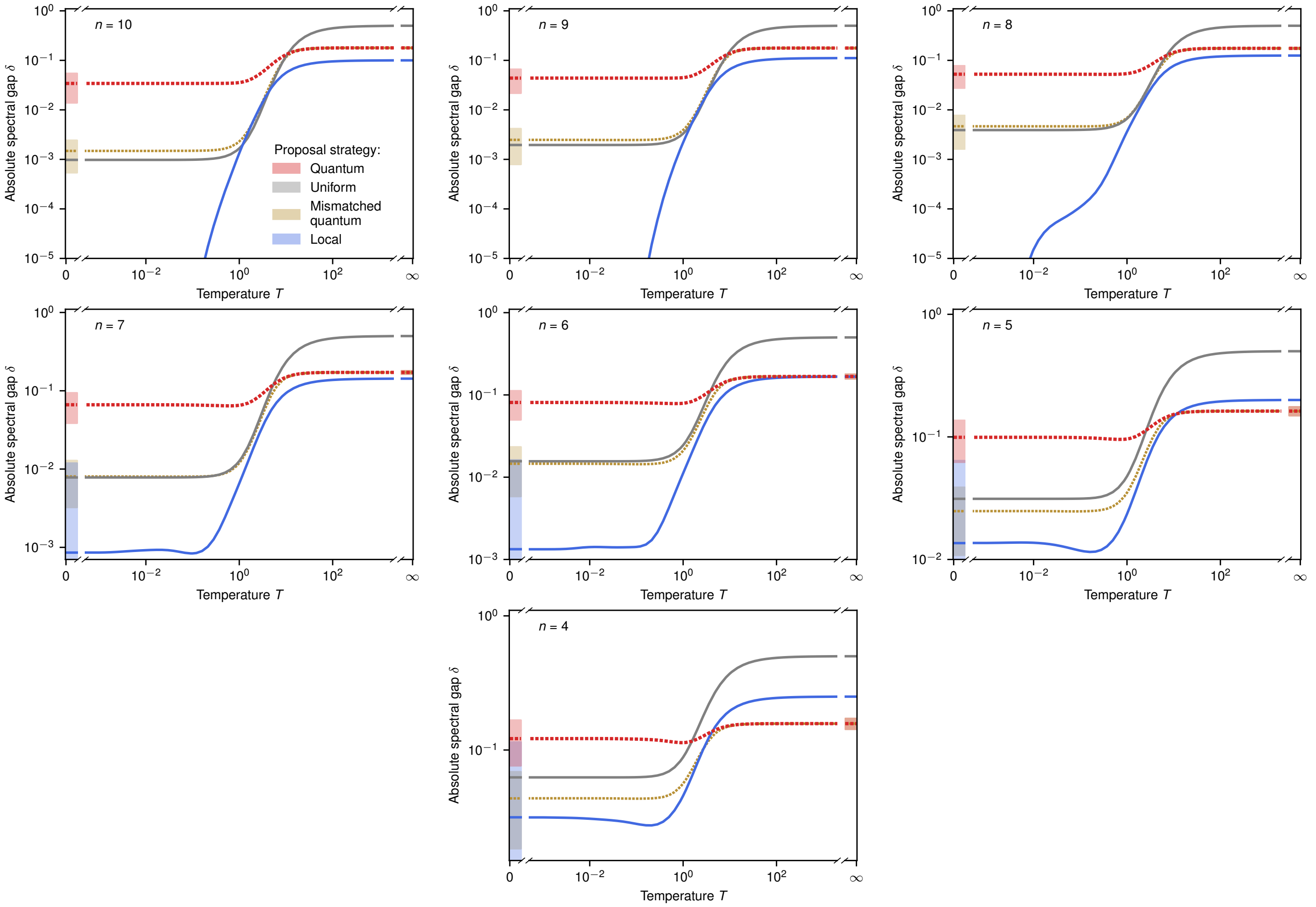


FIG. S4. **Average-case Gibbs sampler convergence rate simulations for different n.** Analogous plots to Fig. S2 using the Gibbs sampler acceptance probability A_{G} from Eq. (S8), rather than the Metropolis-Hastings one from Eq. (3) of the main text. All strategies were simulated classically. Lines show the average δ over 500 random fully-connected Ising model instances for each n; error bands show the standard deviation in δ over these instances. Dotted lines are for visibility. The same model instances were used to make Fig. 2 of the main text.

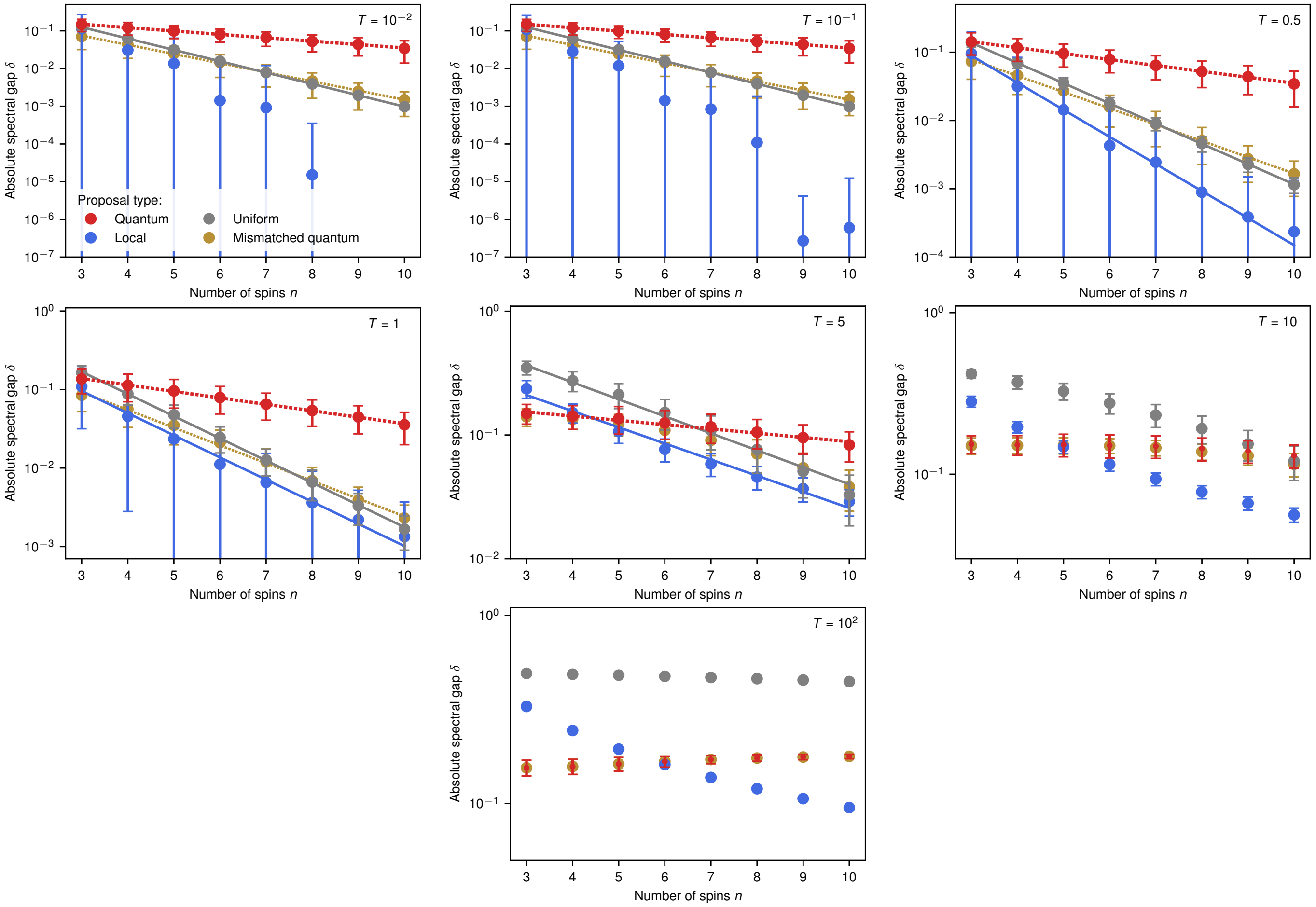


FIG. S5. **Average-case Gibbs sampler convergence rate simulations for different T.** Analogous plots to Fig. S3 using the Gibbs sampler acceptance probability A_{G} from Eq. (S8), rather than the Metropolis-Hastings one from Eq. (3) of the main text. All strategies were simulated classically. Markers show the average δ over 500 random fully-connected Ising model instances for each n; error bars show the standard deviation in δ over these instances. Lines show least squares exponential fits $\langle\delta\rangle \propto 2^{-kn}$, weighted by the standard error of the mean, wherever such fits are reasonably good. The resulting values of k are given in Table S2. Dotted lines are for visibility. The same model instances were used to make Fig. 2 of the main text.

Proposal type	$T = 10^{-2}$	$T = 10^{-1}$	$T = 0.5$	$T = 1$	$T = 5$	$T = 10$
Quantum	$k = 0.299(3)$	$k = 0.298(3)$	$k = 0.288(3)$	$k = 0.273(4)$	$k = 0.114(8)$	-
Mismatched quantum	$k = 0.806(9)$	$k = 0.803(9)$	$k = 0.79(1)$	$k = 0.75(1)$	-	-
Local	-	-	$k = 1.32(4)$	$\boldsymbol{k = 0.94(4)}$	$\boldsymbol{k = 0.43(2)}$	-
Uniform	$\boldsymbol{k = 1}$	$\boldsymbol{k = 0.9989(4)}$	$\boldsymbol{k = 0.983(2)}$	$k = 0.940(7)$	$k = 0.46(2)$	-
Quantum enhancement	3.34(3)	3.35(3)	3.42(3)	3.4(1)	3.8(3)	-

TABLE S2. **Gibbs sampler $\langle\delta\rangle$ vs. n fits on fully-connected instances.** The exponential fit parameters k from $\langle\delta\rangle \propto 2^{-kn}$ in Fig. S5 for different proposal strategies, wherever such fits are reasonably good. The resulting average quantum enhancement exponent for each T, which is the ratio of k for the quantum algorithm and the smallest k among classical proposal strategies (shown in bold), is given in the bottom row. The parameters here are analogous to those in Table S1, but for the Gibbs sampler acceptance probability A_{G} from Eq. (S8) rather than the Metropolis-Hastings one from Eq. (3) of the main text.

4. Lazy chains

Section III C 3 highlighted the possibility that δ can be small because the MCMC transition matrix $\boldsymbol{P}$ has an eigenvalue close to -1. (This effect occurs for the local proposal at high T in Figs. 2a and 3 of the main text.) While this indeed reflects slow convergence, it can be remedied in a trivial way by using a lazy Markov chain. Suppose $\boldsymbol{P}$ is the transition matrix for a Markov chain; the lazy counterpart of this chain has transition matrix $\boldsymbol{P}_{\text{lazy}} = \frac{1}{2}(\boldsymbol{P} + \boldsymbol{I})$. To realize the lazy chain, at each step one either jumps according to $\boldsymbol{P}$ or stays at the current configuration, each with 50% probability. Perhaps surprisingly, this lazy chain can converge faster than the regular chain. If $\boldsymbol{P}$ has eigenvalues $\{\lambda\}$, then $\boldsymbol{P}_{\text{lazy}}$ has eigenvalues $\{\frac{\lambda+1}{2}\}$. So if an eigenvalue of $\lambda \approx -1$ limits δ in the former, it typically does not in the latter, resulting in faster convergence.

Given the trivial nature of the speedup that can be obtained through a lazy chain, it is essential to check that the quantum enhancement in δ we observed is not due to a similar effect, owing to an eigenvalue near -1. Indeed, it is not, as shown in Figs. S6 and S7, and Table S3. In fact, the only setting where lazy chains enhance δ is for the local proposal at high T. Otherwise, they only serve to slightly reduce δ.

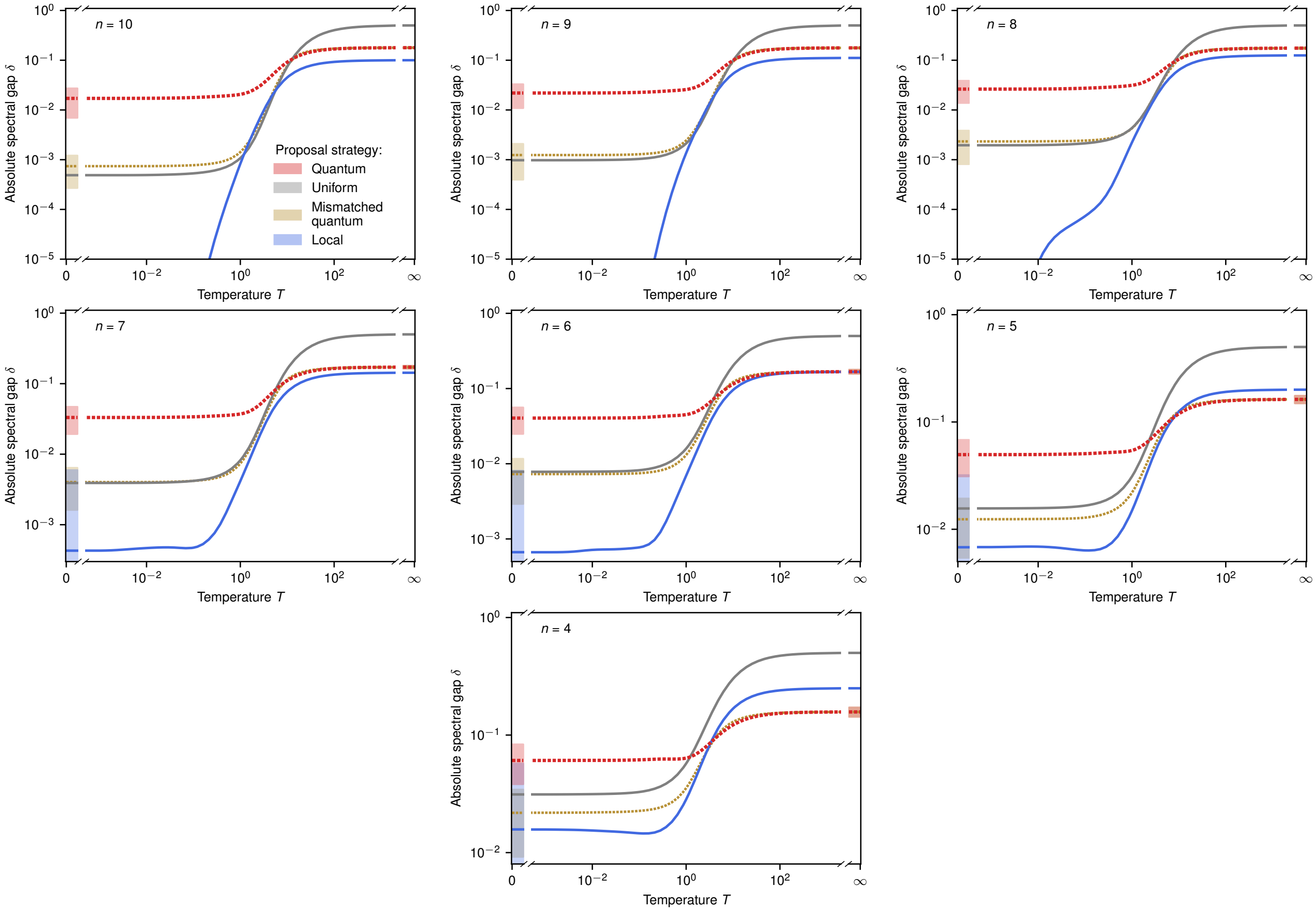


FIG. S6. **Average-case lazy M-H convergence rate simulations for different n.** Analogous plots to Fig. S2 using lazy Metropolis-Hastings chains. All strategies were simulated classically. Lines show the average δ over 500 random fully-connected Ising model instances for each n; error bands show the standard deviation in δ over these instances. Dotted lines are for visibility. The same model instances were used to make Fig. 2 of the main text.

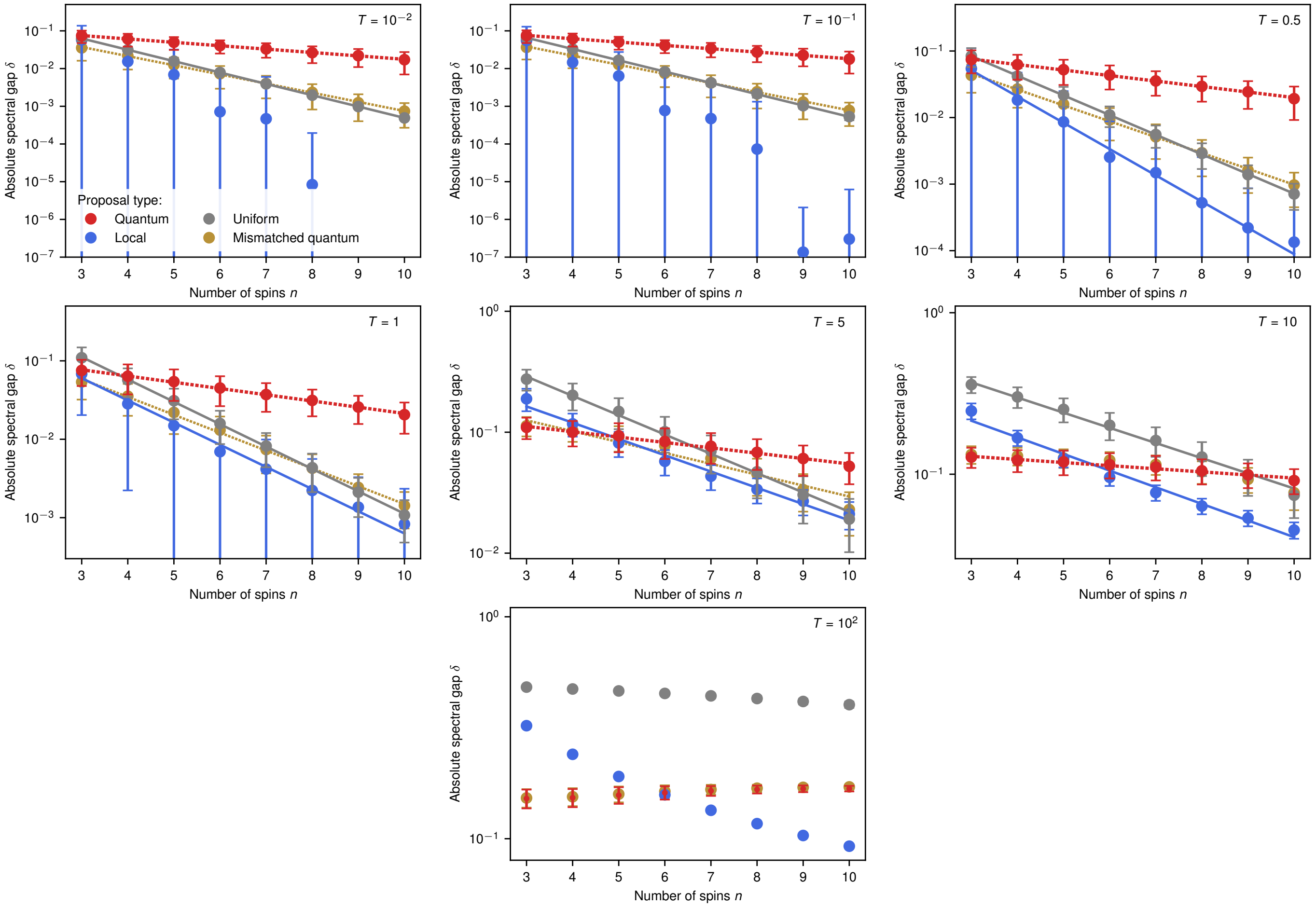


FIG. S7. **Average-case lazy M-H convergence rate simulations for different T.** Analogous plots to Fig. S3 using lazy Metropolis-Hastings chains. All strategies were simulated classically. Markers show the average δ over 500 random fully-connected Ising model instances for each n; error bars show the standard deviation in δ over these instances. Lines show least squares exponential fits $\langle\delta\rangle \propto 2^{-kn}$, weighted by the standard error of the mean, wherever such fits are reasonably good. The resulting values of k are given in Table S3. Dotted lines are for visibility. The same model instances were used to make Fig. 2 of the main text.

Proposal type	$T = 10^{-2}$	$T = 10^{-1}$	$T = 0.5$	$T = 1$	$T = 5$	$T = 10$
Quantum	$k = 0.299(3)$	$k = 0.293(2)$	$k = 0.275(4)$	$k = 0.264(4)$	$k = 0.147(6)$	$k = 0.064(4)$
Mismatched quantum	$k = 0.806(9)$	$k = 0.803(8)$	$k = 0.792(9)$	$k = 0.76(1)$	$k = 0.30(3)$	-
Local	-	-	$k = 1.31(4)$	$\boldsymbol{k = 0.94(4)}$	$\boldsymbol{k = 0.45(2)}$	$k = 0.34(2)$
Uniform	$\boldsymbol{k = 0.9999(2)}$	$\boldsymbol{k = 0.995(3)}$	$\boldsymbol{k = 0.981(4)}$	$k = 0.948(7)$	$k = 0.53(2)$	$\boldsymbol{k = 0.31(1)}$
Quantum enhancement	3.35(3)	3.39(3)	3.57(6)	3.6(1)	3.0(2)	4.9(4)

TABLE S3. **Lazy M-H $\langle\delta\rangle$ vs. n fits on fully-connected instances.** The exponential fit parameters k from $\langle\delta\rangle \propto 2^{-kn}$ in Fig. S7 for different proposal strategies, wherever such fits are reasonably good. The resulting average quantum enhancement exponent for each T, which is the ratio of k for the quantum algorithm and the smallest k among classical proposal strategies (shown in bold), is given in the bottom row. The parameters here are analogous to those in Table S1, but for lazy Metropolis-Hastings chains.

D. Cluster Algorithms

In Section IV we show additional experimental data similar to Fig. 3 of the main text. (Namely, Figs. S16, S21 and S28.) For completeness, we also computed δ for five common MCMC cluster algorithms (which are the most common ones, to the best of our knowledge) in these supplementary plots. One would not expect them to converge particularly fast, since none of them are intended for fully-connected or 1D spin glasses. Indeed we found their spectral gaps to be comparable with those of the local and uniform proposals, despite the underlying algorithms being substantially more complicated, both conceptually and practically.

1. Swendsen-Wang and Wolff clusters

The Swendsen-Wang (S-W) and Wolff MCMC algorithms work by grouping spins into clusters based on the model instance, the current configuration $\boldsymbol{s}$, and the temperature T [9, 10]. (See Refs. [11, 12] for spin glass implementations.) In the S-W algorithm multiple clusters can then be flipped (flipping a cluster means $s_j \mapsto -s_j$ for each spin j in the cluster), whereas in the Wolff algorithm only a single one can. We are aware of two main variants (which are in fact distinct algorithms) for both the S-W and Wolff algorithms, which differ in how the fields $\{h_j\}$ are handled. The first uses an additional "ghost spin," whose state is held fixed, to turn the fields into couplings as discussed in Section III B [13]. The other uses an additional accept/reject step that depends only on $\{h_j\}$ and T, unlike the more common accept/reject probabilities discussed in the main text [11, 14, 15]. These two versions of the S-W and Wolff algorithms account for four of the five cluster algorithms considered here. We are not aware of any closed-form expression for the transition probabilities $P(\boldsymbol{s'}|\boldsymbol{s})$ for any of these four algorithms. Rather, we estimated each $P(\boldsymbol{s'}|\boldsymbol{s})$ by

$$\hat{P}(\boldsymbol{s'}|\boldsymbol{s}) = \frac{\text{number of } \boldsymbol{s} \to \boldsymbol{s'} \text{ transitions observed}}{\text{number of } \boldsymbol{s} \to [\text{anything}] \text{ transitions observed}}, \tag{S9}$$

where initial states $\boldsymbol{s}$ were chosen uniformly at random in each repetition. Finally, we formed $\{\hat{P}(\boldsymbol{s'}|\boldsymbol{s})\}$ into a $2^n \times 2^n$ stochastic matrix $\hat{\boldsymbol{P}}$, which we diagonalized to yield an estimator $\hat{\delta}$ for the true absolute spectral gap δ as described in the main text. Finally, we repeatedly resampled the data, where each data point consists of an observed $\boldsymbol{s} \to \boldsymbol{s'}$ transition, to compute bootstrap confidence intervals (CIs) for δ. We picked the initial states $\boldsymbol{s}$ uniformly at random when gathering data (rather than iterating over $\boldsymbol{s} \in \{-1, 1\}^n$ in some fixed order) so that the data points would be IID, which simplifies the computation of bootstrap CIs.

The resulting absolute spectral gaps are shown in Figs. S16, S21 and S28 for illustrative model instances, together with experimental data for our quantum algorithm. We did not repeat this analysis on large ensembles of instances as in Fig. 2 of the main text, however, for three main reasons. First, the analysis is computationally intensive, especially since the cluster formation probabilities depend on T, so this sampling procedure must be repeated not just for each instance, but also for each T. Second, this approach inevitably entails some uncertainty in δ, which is hard to control in an automated way, and makes it complicated to average δ over many model instances. Indeed, estimating δ confidence intervals for a single instance through bootstrapping (which must be done separately for each value of T considered) is, in itself, quite computationally intensive. Finally, these algorithms are not designed for spin glasses. The mediocre absolute spectral gaps we observed as a result on illustrative model instances did not justify the considerable computational cost required to analyze them over many such instances.

Note finally that the S-W and Wolff algorithms (even the versions with an accept/reject step) do not fit into the simple framework described in the main text wherein a jump $\boldsymbol{s} \to \boldsymbol{s'}$ is proposed with some T-independent probability $Q(\boldsymbol{s'}|\boldsymbol{s})$ and then accepted with some T-dependent probability $A(\boldsymbol{s'}|\boldsymbol{s})$, since the cluster formation process depends on T. This extra temperature dependence prevents us from showing these cluster algorithms in Fig. 5 of the main text, and the like, as there is no obvious quantity for these algorithms admitting a fair comparison to $Q(\boldsymbol{s'}|\boldsymbol{s})$ for the local, uniform and quantum proposals.

2. *Houdayer clusters*

The other MCMC cluster algorithm we considered is that of Houdayer [16]. Analyzing its performance poses two main difficulties: First, finding δ can be computationally intensive. Second, the algorithm uses more resources than those considered in the main text, making it difficult to achieve a fair comparison. For one, Houdayer's algorithm uses $R \geq 2$ independent replicas of an Ising model instance simultaneously at the same T. (Here, "replicas" refer to multiple model instances with the same couplings $\{J_{jk}\}$ and fields $\{h_j\}$ but in independent spin configurations, i.e., the simulation cell consists of nR total spins.) In each iteration, the replicas are paired together and their states are compared in such a way as to form clusters, which are then flipped. To achieve the fairest possible comparison with the other algorithms we considered, all of which use a single replica, we focus on the simplest case of $R = 2$ replicas.

Other issues hindering fair comparison arise immediately. For instance, unlike in the S-W and Wolff algorithms, this cluster flipping procedure alone does not guarantee convergence. Rather, it is meant to be paired with n local Metropolis-Hastings jumps on each cluster. (In fact, in each replica, each spin $j \in [1, n]$ is meant to be flipped sequentially, with an accept/reject step after each. This is different than the local proposal described in the main text, where a random spin is picked each time, and does not constitute a Markov chain in the same way. However, the ultimate effect is likely very similar to performing n local Metropolis-Hastings jumps on each replica [17], so we will treat the two as equivalent.) This means that every Houdayer iteration, as described so far, comprises $nR + 1$ jumps. To allow a fair comparison, we instead consider a variant of the algorithm wherein each iteration performs a local Metropolis-Hastings jump on all replicas with probability $n/(n+1)$, or a cluster flip with probability $1/(n+1)$. This way, the average behavior is that of the standard Houdayer algorithm, but only a single operation per cluster is performed in every iteration.

Finally, these steps are often wrapped in a parallel tempering scheme, i.e., they are performed for many different values of T in parallel, and the replicas at different temperatures are made to interact periodically. However, as discussed in the main text, our quantum algorithm could also be wrapped in similar ways, although such generalizations are beyond the scope of this initial work. To allow a fair comparison, then, we use only two replicas at a single temperature T. (Note that this restriction to a single T precludes comparison with another cluster algorithm called isoenergetic cluster moves [18].)

The MCMC transition matrix $\boldsymbol{P}$ for this implementation of the Houdayer algorithm can be constructed exactly, without resorting to sampling of the sort described in Section III D 1. However, even for the simplest case of $R = 2$ replicas, $\boldsymbol{P}$ is $2^{2n} \times 2^{2n}$, so for an $n = 10$ instance it has $2^{40} \approx 10^{12}$ matrix elements. Fortunately it is sparse for the model instances we considered, so δ can be found using an iterative eigenvalue algorithm. Still, doing so up to $n = 10$ is computationally expensive, and difficult to automate. Moreover, the resulting δ follows that of the local proposal closely or exactly. (This was expected, as the algorithm is designed for 2D models, not 1D or fully-connected ones.) So as with the S-W and Wolff algorithms, we analyzed the Houdayer algorithm only on illustrative model instances, rather than for large ensembles of random instances.

We found δ using the `sparse.linalg.eigs` function from the SciPy library [19], which finds the largest eigenvalues in absolute value of a matrix. For $\boldsymbol{P}$ these are $\lambda_1 = 1$ and λ_2, where the latter sets the absolute spectral gap as $\delta = 1 - |\lambda_2| > 0$. Unfortunately, the underlying eigenvalue algorithm can be slow or inaccurate when applied directly to $\boldsymbol{P}$ with $|\lambda_2| \approx 1$, as the near-degeneracy makes it hard to resolve these eigenvalues from each other. We instead preconditioned the problem by first constructing a new linear operator related to $\boldsymbol{P}$ but without this near-degeneracy, allowing us to find δ faster and more reliably. Suppose $\boldsymbol{P}$ is the transition matrix for a Markov chain satisfying detailed balance with a unique stationary distribution $\vec{p} = \boldsymbol{P}\vec{p}$ whose entries are all positive. Then the matrix

$$\boldsymbol{L} = \mathrm{diag}(\vec{p}^{\,-1/2})\,\boldsymbol{P}\,\mathrm{diag}(\vec{p}^{\,1/2}) = \boldsymbol{L}^T, \tag{S10}$$

where $\vec{p}^{\,\pm 1/2}$ is defined element-wise, is symmetric [20]. Moreover, since $\boldsymbol{L}$ and $\boldsymbol{P}$ are similar matrices, they have the same eigenvalues. Because $\boldsymbol{L}$ is symmetric, however, its eigenvectors corresponding to different eigenvalues are orthogonal. In particular, $\sqrt{\vec{p}}$ is the $\lambda = 1$ eigenvector of $\boldsymbol{L}$ satisfying $\|\sqrt{\vec{p}}\| = 1$ since $\sum_j p_j = 1$. Subtracting the orthogonal projector $\sqrt{\vec{p}}\sqrt{\vec{p}}^{\,T}$ onto this eigenspace from $\boldsymbol{L}$ gives the matrix

$$\begin{aligned}\tilde{\boldsymbol{L}} &= \boldsymbol{L} - \sqrt{\vec{p}}\sqrt{\vec{p}}^{\,T} \\ &= \mathrm{diag}(\vec{p}^{\,-1/2})\left[\boldsymbol{P} - \boldsymbol{M}\right]\mathrm{diag}(\vec{p}^{\,1/2}),\end{aligned} \tag{S11}$$

where $\boldsymbol{M} = (\vec{p}\,|\cdots|\,\vec{p})$ has the elements of $\vec{p}$ in each column. $\boldsymbol{L}$ and $\tilde{\boldsymbol{L}}$ have all the same eigenvectors, associated with all the same eigenvalues except for one: $\sqrt{\vec{p}}$ is an eigenvector of $\boldsymbol{L}$ with eigenvalue 1 but an eigenvector of $\tilde{\boldsymbol{L}}$ with eigenvalue 0. Finally, since $\tilde{\boldsymbol{L}}$ and $\boldsymbol{P}-\boldsymbol{M}$ are similar matrices, they have the same spectrum. This means that the largest eigenvalue of $\boldsymbol{P}-\boldsymbol{M}$, in absolute value, is λ_2 rather than 1. Applying an iterative eigenvalue algorithm to $\boldsymbol{P}-\boldsymbol{M}$ rather than $\boldsymbol{P}$ therefore avoids the issue of near-degeneracy due to $|\lambda_2| \approx 1$. And while $\boldsymbol{M}$ is a dense $2^{2n} \times 2^{2n}$ matrix, $\boldsymbol{M}\vec{x} = \vec{p}\sum_j x_j$ for any vector $\vec{x}$. One can therefore apply `sparse.linalg.eigs` to the linear operator $\mathcal{P}$ defined by the action

$$\mathcal{P}(\vec{x}) = \boldsymbol{P}\vec{x} - \vec{p}\sum_j x_j, \tag{S12}$$

without having to store $\boldsymbol{M}$ in memory. We used this method to compute δ as a function of T for the Houdayer algorithm, for which $\boldsymbol{P}$ satisfies detailed balance by construction and the stationary distribution is $\vec{p} = \vec{\mu} \otimes \vec{\mu}$. We constructed $\vec{p}$ by first finding $\vec{\mu}$ by diagonalizing the transition matrix of a single-replica Metropolis-Hastings chain. The results are shown in Figs. S16, S21 and S28 for illustrative model instances, together with S-W and Wolff gaps, and experimental data for our quantum algorithm. Despite their relative complexity, these cluster algorithms perform comparably to M-H with a local or uniform proposal, and converge especially slowly in the low-T regime of interest.

IV. EXPERIMENTAL DETAILS

We performed all of the experiments on *ibmq_mumbai*, a quantum processor with 27 fixed-frequency superconducting transmon qubits forming a heavy hexagonal lattice. In this section we describe the quantum circuits that were implemented, how the resulting data was analyzed, and finally, we show additional experimental data similar to Figs. 3-5 of the main text but for other illustrative model instances.

A. Quantum circuits

1. *Trotter circuits and gates*

To simplify the notation, let

$$H_1 = -(1-\gamma)\alpha\sum_{j=1}^{n} h_j Z_j + \gamma\sum_{j=1}^{n} X_j \tag{S13}$$

and

$$H_2 = -(1-\gamma)\,\alpha \sum_{j>k=1}^{n} J_{jk} Z_j Z_k \tag{S14}$$

so that H from Eq. (5) of the main text can be written as $H = H_1 + H_2$, where H_1 and H_2 contain all the 1- and 2-body terms respectively. For any value of γ and t, we approximated the dynamics $U = e^{-iHt}$ by a 2nd-order product formula (PF)

$$V = \left(e^{-iH_2\Delta t/2}\, e^{-iH_1\Delta t}\, e^{-iH_2\Delta t/2}\right)^r, \tag{S15}$$

where $\Delta t = t/r$ for an integer number of timesteps r. The unitary $e^{-iH_1\Delta t}$ can be realized through single-qubit gates applied in parallel, while $e^{-iH_2\Delta t/2}$ is implemented through two layers of 2-qubit gates, as detailed below. In theory, one could equally well exchange $H_1 \leftrightarrow H_2$ in Eq. (S15). However, the order we used requires fewer 2-qubit gates than the alternative. This is because our algorithm only involves initial states and measurements in the computational basis, so the first and last $e^{-iH_2\Delta t/2}$ layers have no impact on the measurement statistics, and therefore need not be implemented at all. Rather, one can implement

$$\begin{aligned}\tilde{V} &= \left(e^{-iH_1\Delta t}\, e^{-iH_2\Delta t/2}\right)\left(e^{-iH_2\Delta t/2}\, e^{-iH_1\Delta t}\, e^{-iH_2\Delta t/2}\right)^{r-2}\left(e^{-iH_2\Delta t/2}\, e^{-iH_1\Delta t}\right) \\ &= e^{-iH_1\Delta t}\left(e^{-iH_2\Delta t}e^{-iH_1\Delta t}\right)^{r-1}\end{aligned} \tag{S16}$$

✓

$r = 3$

$e^{-H_1}\, e^{-H_2/2}\, \left(e^{-iH_2/2} - e^{-iH_1}\, e^{-iH_2/2}\right) \qquad \left(e^{-i\frac{H_2}{2}} e^{-iH_1}\right)$

$e^{-H_1} \left(e^{-iH_2} e^{-iH_1}\right)\left(e^{-iH} e^{-iH_1}\right)$

in place of V, therefore realizing a 2^{nd}-order r-step PF (or equivalently, a 1^{st}-order r-step PF [21]) using only $r-1$ applications of the entangling Trotter step $e^{-iH_2\Delta t}$.

The resulting Trotter circuit is shown in Fig. S8. Each $e^{-iH_1\Delta t}$ step can be implemented by applying $\exp[-i(a_jX_j + b_jZ_j)]$ for $a_j = \gamma\Delta t$ and $b_j = -(1-\gamma)\alpha h_j \Delta t$ on all qubits j in parallel. We realized these 1-qubit unitaries using Qiskit's "U" gates, which are implemented using two X90 pulses along with virtual Z rotations [22]. Each $e^{-iH_2\Delta t}$ step can be decomposed in terms of composite $R_{ZZ}(\theta) = e^{-i\theta/2\, Z\otimes Z}$ operations, with rotation angles of $\theta_{jk} = -2J_{jk}(1-\gamma)\alpha\Delta t$ between qubits j and k. For this initial demonstration, we implemented our algorithm exclusively for 1D problem instances in order to avoid SWAP gates, and so that each $e^{-iH_2\Delta t}$ step could be realized through only two layers of parallel R_{ZZ} rotations, therefore reducing the circuit depth for fixed r.

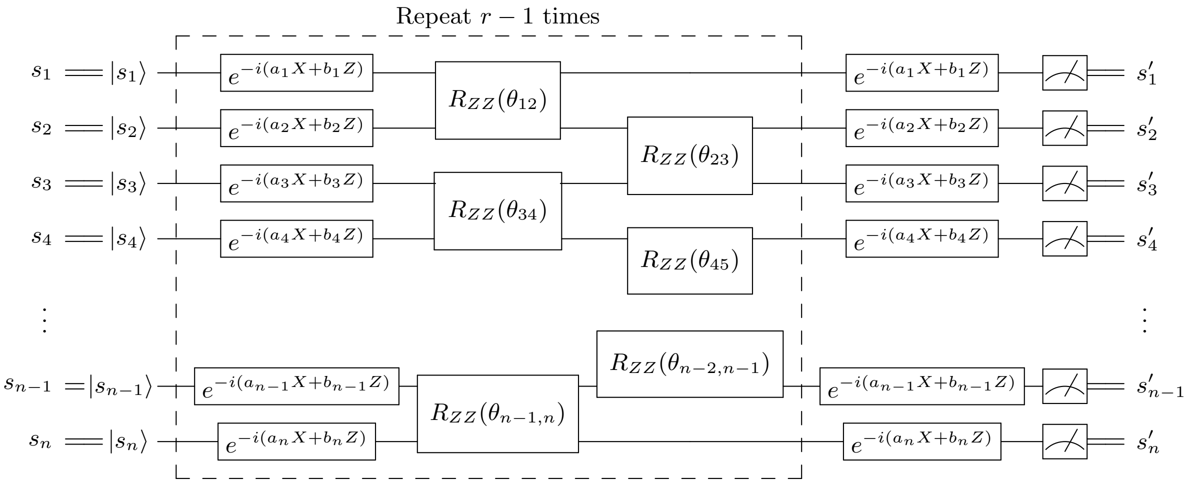


FIG. S8. **Trotter circuit.** A n-qubit circuit approximating e^{-iHt} through an r-step 2^{nd} order product formula. It applies the unitary $\tilde{V}$ from Eq. (S16) to an initial computational state $|\boldsymbol{s}\rangle$, then performs a measurement in the computational basis which yields an n-bit string $\boldsymbol{s'}$.

A common way to realize $R_{ZZ}(\theta)$ rotations is through the double-CNOT decomposition in Fig. S9b, which uses two fully-entangling CNOT gates and a single-qubit rotation $R_Z(\theta) = e^{-i\theta/2\, Z}$ [23]. CNOTs on *ibmq_mumbai* are implemented through an echoed cross-resonance (CR) gate which (ideally) performs an $R_{ZX}(\theta) = e^{-i\theta Z\otimes X/2}$ rotation with $\theta = \pi/2$ using two flat-top CR pulses and rotary tones together with an echo sequence [24, 25]. We instead realized $R_{ZZ}(\theta)$ rotations through the pulse-efficient method of Refs. [26, 27], shown in Fig. S9c and available through Qiskit, which scales the area of the CNOT CR and rotary pulses to implement echoed $R_{ZX}(\theta)$ gates for arbitrary angles θ. This approach was thoroughly characterized in [27], where it led to an error reduction of up to 50% compared to the double-CNOT decomposition due to its substantially shorter pulse schedule. Crucially, since it scales the pulse parameters from calibrated CNOT gates to realize arbitrary angles θ, this pulse-efficient approach requires no additional calibration. Arbitrary angles θ can be folded into the interval $[-\pi, \pi]$ by adding/subtracting integer multiples of 2π since $R_{ZX}(\theta+\phi) = R_{ZX}(\theta)R_{ZX}(\phi)$ and $R_{ZX}(\pm 2\pi) = -I$, as shown in Fig. S10b. They can then be folded further into $[-\pi/2, \pi/2]$ by adding/subtracting π and introducing additional 1-qubit gates since $R_{ZX}(\pm\pi) = \mp iZ \otimes X$, as shown in Fig. S10c. Using this approach, we implemented every $R_{ZZ}(\theta)$ rotation in each $e^{-iH_2\Delta t}$ Trotter step using a pulse-efficient R_{ZX} rotation with an angle in $[-\pi/2, \pi/2]$; that is, with a fraction of a CNOT rather than two CNOTs, together with single-qubit gates. For each R_{ZX} gate, we picked the control (Z) and target (X) qubits based on the direction that gave the fastest CNOT.

On an ideal quantum computer, the Trotter timestep duration Δt could be chosen to ensure that the final state is within some desired error tolerance of the target state given by the Schrödinger equation [28]. On an actual, noisy quantum computer, however, one can only hope to pick Δt (or equivalently, the number of timesteps r) minimizing the aggregate error from discretizing the target dynamics (i.e., Trotter error) and from gate errors [29–31]. When r is small (and Δt is large) the former type of error dominates, while when r is large (and Δt is small) the latter dominates. The optimal choice of Δt could therefore depend on H, t, and on the nature of the experimental gate errors. We used a Trotter timestep of $\Delta t = 0.8$ for all experiments in this work. We found this value to give a reasonably good trade-off between gate and Trotter errors for the implementation described above, independent of t (as predicted in Ref. [29]), γ, and the model instance. Moreover, it is sufficiently large that all R_{ZX} gates could be

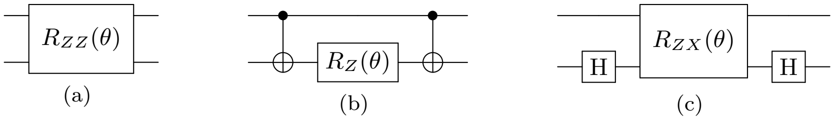


FIG. S9. **Ways to implement $R_{ZZ}(\theta)$.** The target operation $R_{ZZ}(\theta)$ in panel (a) is often implemented using the equivalent circuit in (b), which we call the double-CNOT decomposition. We instead implemented it through the equivalent circuit in (c), where H denotes a Hadamard gate (not the Hamiltonian H).

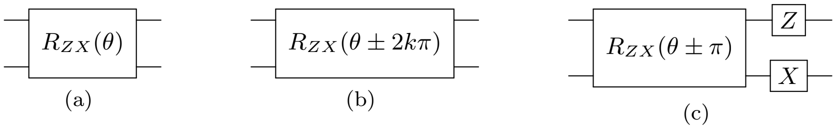


FIG. S10. **Wrapping $R_{ZX}(\theta)$ angles.** Equivalent circuits for arbitrary angles θ and integers k.

implemented by shortening the flat-top portion of CR pulses, rather than scaling their amplitude [27].

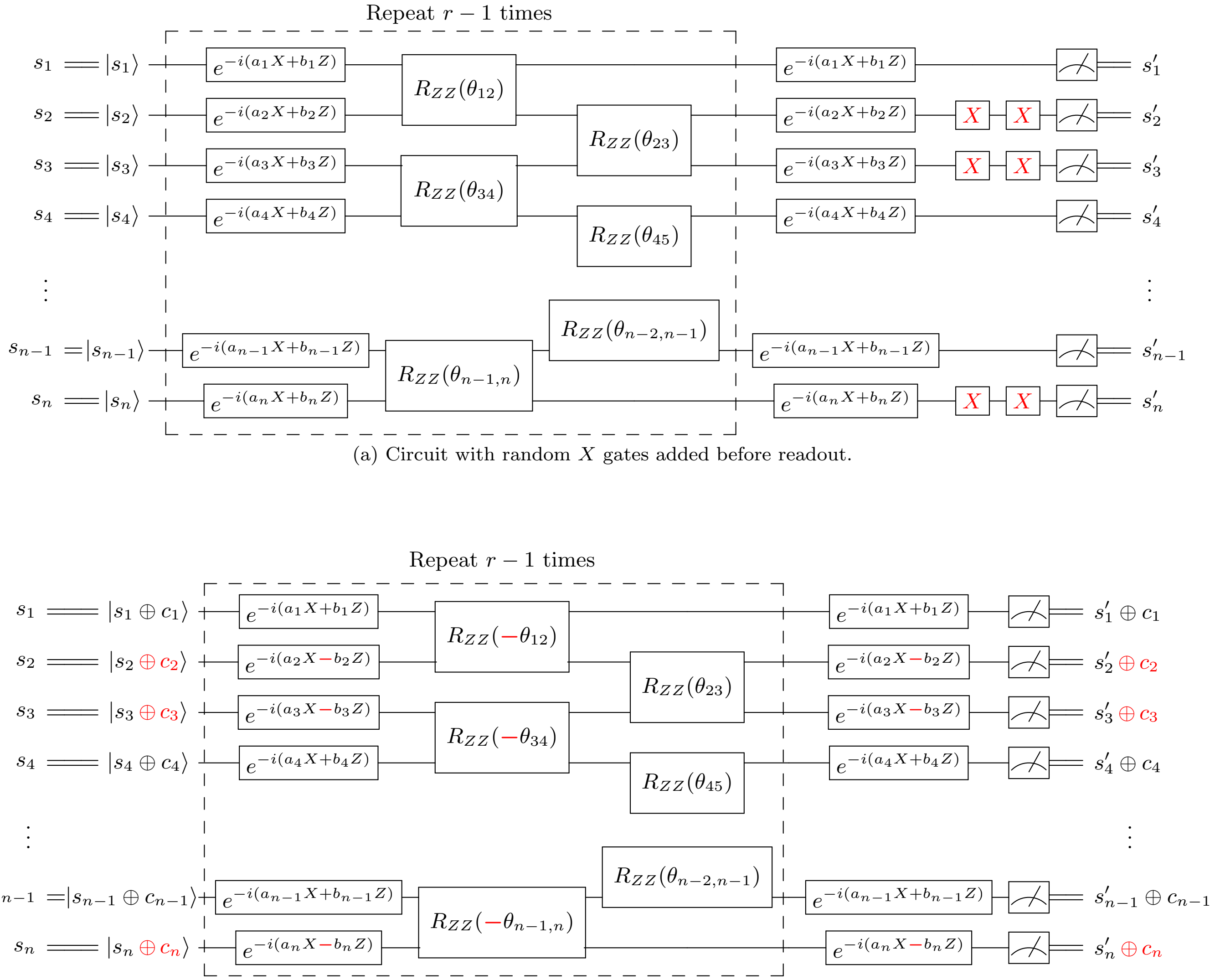
2. Twirling

We sought not only to minimize experimental errors, but also to ensure that any remaining errors satisfied the symmetry requirement $Q(\boldsymbol{s}'|\,\boldsymbol{s}) = Q(\boldsymbol{s}\,|\boldsymbol{s}')$ for our algorithm. (We emphasize that quantum errors do not necessarily bias our MCMC algorithm, i.e., cause it to converge to the wrong distribution. Rather, it is asymmetry in the quantum errors that can have this effect.) We found that implementing bare Trotter circuits like that in Fig. S8 clearly breaks this symmetry, which could bias our algorithm. We attribute this to biased errors in the two-qubit gates and the readout, which we expect to be the noisiest components of our experiments. By randomizing the errors in both we were able to restore the necessary symmetry.

To randomize the readout errors, we picked a subset of qubits uniformly at random and—in principle—added two adjacent X gates (which is equivalent to the identity gate) to each of those qubits before the readout, as shown in Fig. S11a. For each selected qubit, we commuted one X gate through the measurement, transforming it into a classical NOT gate on the output bit. This removes any preferred direction in the readout errors on each qubit (e.g., if a $1 \to 0$ error was more likely to occur than a $0 \to 1$ error). Rather than physically implementing the remaining X gates, we chose to commute them through the Trotter unitaries and into the initial state, in the spirit of [32], as shown in Fig. S11b. This leaves the circuit's structure unchanged, but maps $Z_j \mapsto -Z_j$ for all selected qubits j. The net effect is to randomize both the state preparation and measurement (SPAM) errors—accordingly, we refer to this procedure as *SPAM twirling*. We will describe which qubits get selected by an n-bit *key* $\boldsymbol{c}$. In keeping with our convention of $\boldsymbol{s} \in \{-1,1\}^n$ rather than $\{0,1\}^n$ for the Ising model, we take $\boldsymbol{c} \in \{-1,1\}^n$ and $c_j = -1$ to mean qubit j is selected ($c_j = 1$ otherwise). We use $\boldsymbol{s} \oplus \boldsymbol{c}$ to denote bitwise XOR, which in our notation is simply $(s_1c_1, \ldots, s_nc_n)$. The above procedure of adding random X gates is logically equivalent to:

1. picking a random key $\boldsymbol{c} \in \text{uniform}(\{-1,1\}^n)$,
2. preparing an initial state $|\boldsymbol{s} \oplus \boldsymbol{c}\rangle$ in place of $|\boldsymbol{s}\rangle$
3. implementing a Trotter circuit with $Z_j \mapsto -Z_j$ for all j where $c_j = -1$, or equivalently, $J_{jk} \mapsto J_{jk}c_jc_k$ and $h_j \mapsto h_jc_j$,
4. applying $\boldsymbol{c} \oplus \cdot$ to the measured bit string (i.e., XOR-ing it with $\boldsymbol{c}$) to recover the intended output $\boldsymbol{s}'$.

This is the procedure we implemented in all of the experiments presented here. On an ideal quantum computer, it would yield identical measurement statistics (for $\boldsymbol{s}'$, as a function of $\boldsymbol{s}$) to the bare circuit in Fig. S8 for any key $\boldsymbol{c}$. (Note though that Figs. S8 and S11b do not implement the same unitary; rather, they produce equivalent quantum transition probabilities after post-processing the input and output strings. For brevity, we will nonetheless call the circuits equivalent.) On a noisy quantum computer, however, some readout errors (e.g., T_1 errors during measurement) may be more likely than others. But the effect of these more-likely errors on $\boldsymbol{s'}$ depends on which key $\boldsymbol{c}$ is used. Using many random keys (together with the 2-qubit gate twirling described below) removed any detectable asymmetry from our experimental results, as detailed in Section IV C.



(a) Circuit with random X gates added before readout.

(b) Circuit with the random X gates absorbed into the initial and final states.

FIG. S11. **SPAM twirling.** The top circuit (a) is equivalent to the bare Trotter circuit in Fig. S8, but has pairs of X gates added to random qubits (qubits 2, 3 and n in this example, shown in red). To produce the bottom circuit (b), one X gate from each pair is commuted through $\tilde{V}$ and into the initial state, while the other is moved past the readout. The impact is highlighted in red. The bottom circuit produces identical statistics to the top one in theory, provided one prepares the initial state $|\boldsymbol{s} \oplus \boldsymbol{c}\rangle$ for a current MCMC configuration $\boldsymbol{s}$ and XORs the measured string (denoted $\boldsymbol{s'} \oplus \boldsymbol{c}$) with $\boldsymbol{c}$ to recover $\boldsymbol{s'} = \boldsymbol{c} \oplus (\boldsymbol{s'} \oplus \boldsymbol{c})$. In this example $\boldsymbol{c} = (1, -1, -1, 1, \ldots, 1, -1)$.

We also randomized every $R_{ZX}(\theta)$ gate independently as shown in Fig. S12, inspired by Refs. [33–35]. That is, for each occurrence we picked IID random single-qubit Paulis $P_1, P_2 \sim \text{uniform}(\{I, X, Y, Z\})$ to add before R_{ZX}, as in Fig. S12b. We then sought to undo the effect of these additional Paulis after the R_{ZX} gate. However, because $R_{ZX}(\theta)$ is not generally a Clifford gate, it is not always possible to find Paulis (nor single-qubit gates, more generally) P_3 and P_4 such that $R_{ZX}(\theta) = (P_3 \otimes P_4) R_{ZX}(\theta)(P_1 \otimes P_2)$. Rather, if $[P_1 \otimes P_2, Z \otimes X] = 0$ we undid these Paulis by re-applying them after R_{ZX} (i.e., we took $P_3 = P_1$ and $P_4 = P_2$), as in Fig. S12b. Otherwise, if $\{P_1 \otimes P_2, Z \otimes X\} = 0$, we did the same thing but with $\theta \mapsto -\theta$, as shown in Fig. S12c. Finally, we consolidated all adjacent 1-qubit gates on each qubit into single gates before executing the circuit. We refer to this procedure, informally, as *gate twirling*. (Informally, because the two-qubit gate we apply depends on the choice of P_1 and P_2, so the process may or may not be considered "twirling," depending on one's definition.) Like SPAM twirling, the motivation for this procedure is to remove any preferred direction from the noise on average—here in the $R_{ZX}(\theta)$ gates rather than the readout. Unlike SPAM twirling, however, gate twirling does not change the unitary described by the circuit. We used both SPAM and gate twirling in all of the experiments presented here.

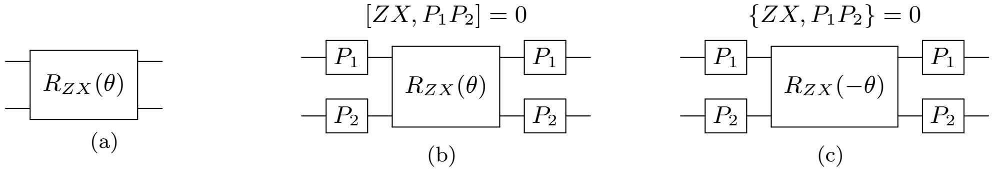


FIG. S12. **Gate twirling.** Rather than implement the target gate $R_{ZX}(\theta)$ in panel (a) directly, we drew Paulis P_1 and P_2 uniformly at random, independently. Depending on whether $P_1 \otimes P_2$ commuted or anti-commuted with $Z \otimes X$, we implemented the equivalent circuits (b) or (c) respectively.

3. Overall structure of experiments

All of the experiments in this work consisted of repeatedly preparing a computational state $|\boldsymbol{s}\rangle$, implementing a twirled Trotter circuit, and counting the measured computational states $|\boldsymbol{s'}\rangle$. Combining such counts for different circuits gave estimates for all the proposal probabilities $Q(\boldsymbol{s'}|\,\boldsymbol{s})$, as shown in Fig. 5b of the main text and similar figures in Section IV C, which we then used to estimate $P(\boldsymbol{s'}|\boldsymbol{s})$ and δ. We also used the same data to extract MCMC chains, as shown in Fig. 4 of the main text and the like in Section IV C, using a resampling procedure described in Section IV B. In principle, we could have collected such data by iterating over all 2^n initial states $|\boldsymbol{s}\rangle$ for each circuit and recorded measurement counts for each. In practice, however, it was more experimentally convenient to prepare initial states uniformly at random by preparing a $|+\rangle^{\otimes n} = 2^{-n/2}\sum_{\boldsymbol{s}}|\boldsymbol{s}\rangle$ state, collapsing it to a random $|\boldsymbol{s}\rangle$ through a first measurement, then applying a Trotter circuit followed by a second measurement, as shown in Fig. S13a. This approach greatly reduces the time spent loading circuits into the classical control hardware, but increases the number of shots required to observe a fixed number of quantum transitions. The time required to initialize the classical control hardware versus that required for each additional shot makes this a favorable trade-off.

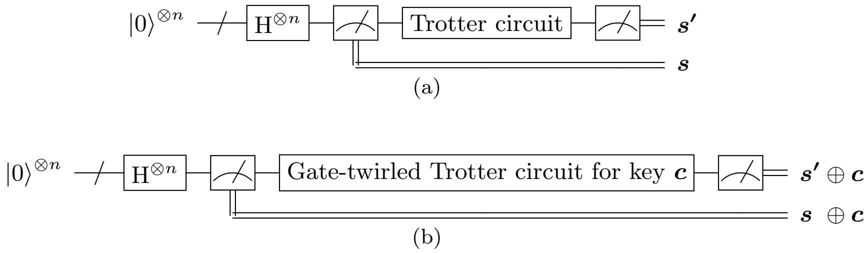


FIG. S13. **Overall structure of experiments.** The top circuit (a) represents the logical structure of the experiments, where "Trotter circuit" refers to the unitary in Fig. S8 and H denotes a Hadamard gate (not the Hamiltonian H). Running this circuit repeatedly samples quantum transitions $|\boldsymbol{s}\rangle \to |\boldsymbol{s'}\rangle$ according to $\Pr(\boldsymbol{s}) = 2^{-n}$ and $\Pr(\boldsymbol{s'}|\,\boldsymbol{s}) = \left|\langle\boldsymbol{s'}|V|\boldsymbol{s}\rangle\right|^2$. The bottom circuit (b) is the one we implemented experimentally. It is logically equivalent to (a), but uses SPAM twirling with a random key $\boldsymbol{c}$ as in Fig. S11b, and gate twirling on all the resulting R_{ZX} gates independently, as in Fig. S12. We then XOR-ed both classical registers with $\boldsymbol{c}$ to recover the intended $\boldsymbol{s}$ and $\boldsymbol{s'}$.

To further reduce the time required to collect data, we considered a close variant of Algorithm S1 in which both γ and t are picked from discrete, rather than continuous, uniform distributions in lines 4 and 5. For γ, we discretized the interval $[0.25, 0.6]$ into 10 subintervals (half the number used for numerics, cf. Eq. (S3)) and considered $\gamma \in \Gamma$ for

$$\Gamma = \left\{0.25 + \frac{\Delta\gamma}{2},\ 0.25 + \frac{3\Delta\gamma}{2},\ 0.25 + \frac{5\Delta\gamma}{2},\ \ldots,\ 0.6 - \frac{\Delta\gamma}{2}\right\} \tag{S17}$$

using a subinterval width of $\gamma = 0.035$. Similarly, rather than use $t \in [2, 20]$ as in our numerics, we considered only integer values of the Trotter timestep $\Delta t = 0.8$, namely $t \in \mathcal{T}$ for

$$\mathcal{T} = \{r\Delta t \mid 2 \le r \le 25\} = \{1.6, 2.4, 3.2, \ldots, 19.2, 20\}, \tag{S18}$$

where r indicates the number of Trotter steps used for each t. For every $\gamma \in \Gamma$ and $t \in \mathcal{T}$, we implemented N_{twirl} distinct (but equivalent) twirled Trotter circuits with keys $\boldsymbol{c}$ and gate twirls picked uniformly at random. We implemented each such circuit experimentally as shown in Fig. S13b, for N_{shots} shots each. We recorded the number of observed $|\boldsymbol{s}\rangle \to |\boldsymbol{s'}\rangle$ quantum transitions between all computational states $|\boldsymbol{s}\rangle$ and $|\boldsymbol{s'}\rangle$ separately for each circuit, after accounting for the key $\boldsymbol{c}$. We then iterated this procedure over all $\gamma \in \Gamma$ and $t \in \mathcal{T}$. We therefore ran a total of $N_{\text{circ}} \equiv |\Gamma| \times |\mathcal{T}| \times N_{\text{twirl}} = 10 \times 24 \times N_{\text{twirl}}$ different circuits per experiment, and observed a total of $N_{\text{circ}} \times N_{\text{shots}}$ quantum transitions. We picked N_{twirl} and N_{shots} differently based on the number of qubits n, as given in Section IV C.

B. Data analysis

We stored the data from each experiment in a 3-dimensional array C with dimensions $2^n \times 2^n \times N_{\text{circ}}$, where the first, second and third dimensions encode the final states, initial states, and the different circuits respectively. That is, for integers $k, j \in [0, 2^n - 1]$ encoding the initial and final states $|\boldsymbol{s}\rangle = |k\rangle$ and $|\boldsymbol{s'}\rangle = |j\rangle$ respectively, and for the ℓ^{th} Trotter circuit out of N_{circ}:

$$C[j,k,\ell] = \text{number of } |k\rangle \to |j\rangle \text{ quantum transitions observed for the } \ell^{\text{th}} \text{ Trotter circuit.} \tag{S19}$$

We estimated the experimental proposal probabilities $Q(j|k)$ by combining the counts from all circuits; that is, using the estimator

$$\hat{Q}(j|k) = \frac{\sum_{\ell=1}^{N_{\text{circ}}} C[j,k,\ell]}{\sum_{j=0}^{2^n-1} \sum_{\ell=1}^{N_{\text{circ}}} C[j,k,\ell]}. \tag{S20}$$

The resulting matrix $\hat{\boldsymbol{Q}}$ is shown explicitly in Fig. 5b and was used to construct the cumulative histograms for our experiment in Fig. 5c-d, all for the $n = 10$ instance in the main text. The same approach was used for the corresponding figures in Section IV C. We then used $\hat{\boldsymbol{Q}}$ to construct an estimator $\hat{P}(j|k)$ for $P(j|k)$ as

$$\hat{P}(j|k) = \begin{cases} A(j|k)\,\hat{Q}(j|k) & j \neq k \\ 1 - \sum_{\substack{m=1 \\ m \neq j}}^{2^n-1} \hat{P}(j|m) & j = k \end{cases}, \tag{S21}$$

where

$$A(j|k) = \min\left(1, \frac{\mu(j)}{\mu(k)}\right) \tag{S22}$$

is the Metropolis-Hastings acceptance probability from Eq. (3) of the main text for a symmetric proposal strategy. Finally, denoting the eigenvalues of the corresponding matrix $\hat{\boldsymbol{P}}$ (which is left-stochastic, by construction) as $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_{2^n}$ where $1 = |\hat{\lambda}_1| \geq |\hat{\lambda}_2| \geq \cdots \geq |\hat{\lambda}_{2^n}|$, we estimated the absolute spectral gap δ by $\hat{\delta} = 1 - |\hat{\lambda}_2|$. This point estimate, as a function of the temperature T, is shown as a solid red line in Fig. 3 of the main text. The same approach was used for the corresponding figures in Section IV C.

1. *MCMC trajectories*

We did not experimentally implement our MCMC algorithm "online" by alternating between quantum and classical steps in real time. Doing so would have been impractically slow due to limitations in the classical infrastructure around our quantum processor. The dynamic circuit capabilities required for an online implementation are under development [36], and some components have been demonstrated in Ref. [37]. Rather, we generated the MCMC trajectories in Fig. 4 of the main text, and the corresponding figures in Section IV C, by running our algorithm "offline." That is, we first formed a cache—namely, the array C described above—of observed quantum transitions by executing circuits as described above and depicted in Fig. S13b. We then iterated through the steps of our algorithm, but rather than getting proposed jumps $\boldsymbol{s} \to \boldsymbol{s'}$ from the quantum processor in real time, we drew them at random without replacement from the cache. This means that we extracted MCMC trajectories by subsampling the same data used to estimate Q, P and δ above.

In effect, for every circuit ℓ and every state k we formed a list $L_{\ell k}$ of experimentally observed $|k\rangle \to |j\rangle$ transitions repeated by multiplicity. E.g., for circuit $\ell = 2$ and state $k = 1$, if we observed $|k\rangle \to |0\rangle$ three times, $|k\rangle \to |4\rangle$ once, $|k\rangle \to |5\rangle$ twice, and no other $|k\rangle \to$ [anything] transitions, the corresponding list would be $L_{\ell=2,k=1} = [0, 0, 0, 4, 5, 5]$ or some permutation thereof. (The order does not matter.) We then ran our algorithm offline by picking an initial state k uniformly at random, a random circuit index $\ell \sim \text{uniform}(\{1, \ldots, N_{\text{circ}}\})$ IID in each iteration (with replacement), and then drawing a proposed jump $j \sim \text{uniform}(L_{\ell,k})$ without replacement. For instance, if we drew $\ell = 2$ and $k = 1$, we would then pick j uniformly from $L_{\ell=2,k=1} = [0, 0, 0, 4, 5, 5]$ (i.e., with $\Pr(j = 0) = 1/2$, $\Pr(j = 4) = 1/6$, $\Pr(j = 5) = 1/3$). If we drew $j = 0$, we would remove it from the list so that $L_{\ell=2,k=1} = [0, 0, 4, 5, 5]$. We then accepted or rejected the $k \to j$ jump as usual, i.e., used $k = j$ to start the next iteration if accepted, otherwise we used the same k again. We repeated this process until we hit an empty list $L_{\ell k}$, meaning we ran out of data (even though not all lists were necessarily empty). We call this process *Markov chain subsampling.* In Fig. 4 of the main text and corresponding ones in Section IV C, we generated MCMC trajectories of 1000 iterations each in this way. We did not reset the cache for each trajectory, i.e., we did not re-use any data. This means that the chains are statistically independent from each other.

Note that this subsampling procedure is *not* equivalent to sampling j from the estimated distributions $[\hat{Q}(j|k)]_{j=0}^{2^n-1}$ or $[\hat{P}(j|k)]_{j=0}^{2^n-1}$ in each iteration. These alternative approaches would only approximately mimic our algorithm. Our approach, on the other hand, involves post-processing but no approximations.

2. Symmetry testing

Our quantum algorithm assumes the symmetry $Q(\boldsymbol{s'}|\,\boldsymbol{s}) = Q(\boldsymbol{s}\,|\boldsymbol{s'})$ for all computational states $\boldsymbol{s}$ and $\boldsymbol{s'}$ so that the classical accept/reject step can be done efficiently. This symmetry is satisfied in theory by our quantum circuits, and we expect the twirling techniques described above to suppress asymmetry which might arise from experimental imperfections. We use a statistical hypothesis test called the Bowker test to check whether our experimental data is consistent with such symmetry. Even if $\boldsymbol{Q}$ were symmetric, one should not expect the number of experimentally observed $|\boldsymbol{s}\rangle \to |\boldsymbol{s'}\rangle$ quantum transitions to exactly equal that of $|\boldsymbol{s'}\rangle \to |\boldsymbol{s}\rangle$ transitions for all $\boldsymbol{s}$ and $\boldsymbol{s'}$, just as one should not expect a fair coin to turn up "heads" exactly half the time. Rather, some asymmetry in the experimental counts is inevitable even if the underlying process is symmetric, due to statistical fluctuations. It is therefore appropriate to use a statistical test to quantify how asymmetric the counts are, and whether that degree of asymmetry is consistent with the underlying Q being symmetric.

The Bowker test [38], which generalizes the well-known McNemar test [39], serves this purpose. Consider N independent data points, where each datum has two attributes, both of which are labelled $1, \ldots, d$. (For instance, a datum could be $(d-2, 3)$.) Form a $d \times d$ matrix $M = (m_{jk})_{j,k=1}^{d}$ where m_{jk} is the number of data points whose first and second attributes equal j and k respectively (so $\sum_{jk} m_{jk} = N$). The Bowker test aims to determine whether the distribution from which the data was drawn is symmetric; that is, whether $\Pr[(j,k)] = \Pr[(k,j)]$ for all attributes $j, k \in \{1, \ldots, d\}$. It takes this symmetry as the null hypothesis, and constructs the test statistic

$$\chi^2 = \sum_{j>k=1}^{d} \frac{(m_{jk} - m_{kj})^2}{m_{jk} + m_{kj}}. \tag{S23}$$

Intuitively, χ^2 is a clear measure of asymmetry. It comprises a sum of non-negative terms quantifying the discrepancy in counts between each pair of bins m_{jk} and m_{kj}. If the bins contain the same number of counts ($m_{jk} = m_{kj}$) they add nothing to the overall statistic. If they are unequal but contain few data points (m_{jk} and m_{kj} are small) they contribute only slightly to χ^2 since the discrepancy could easily be due to shot noise. But if they are unequal and contain many data points, they contribute strongly to χ^2 since this constitutes strong evidence that $\Pr[(j,k)] \neq \Pr[(k,j)]$. Therefore, a small value of χ^2 suggests symmetry while a large value suggests asymmetry in the underlying distribution (where the meaning of "small" and "large" depends on the number of terms $\binom{d}{2}$ in the sum). More formally, under the null hypothesis, χ^2 follows a chi-squared distribution asymptotically with $\binom{d}{2} = d(d-1)/2$ degrees of freedom [38, 40]. Let F be the cumulative distribution function for such a chi-squared distribution, then the p-value

$$p_{\text{sym}} = 1 - F(\chi^2) \tag{S24}$$

answers the following question: If the underlying distribution were symmetric (null hypothesis), what is the probability of getting data that is as asymmetric, or more asymmetric, than my data M? We take $p_{\text{sym}} \leq 1\%$ as the threshold for significance throughout (for consistency with the 99% confidence intervals used throughout), meaning if $p_{\mathrm{sym}} \leq 0.01$ we reject the null hypothesis and conclude that the underlying distribution is asymmetric. Otherwise, the null hypothesis holds, meaning the data is consistent with symmetry.

In our experiments each datum is an observed $|\boldsymbol{s}\rangle \rightarrow |\boldsymbol{s'}\rangle$ quantum transition, k and j represent the initial and final states respectively, and $d = 2^n$. When $N_{\mathrm{circ}} > 1$ (which is the case in all of the experiments presented here), Bowker's test cannot be applied directly to the full set of counts (i.e., we cannot simply take $m_{jk} = \sum_{\ell=1}^{N_{\mathrm{circ}}} C[j,k,\ell]$) because the data points are not IID, as required by the test. The issue is that we recorded quantum transitions for different quantum circuits, but we iterated through these circuits sequentially rather than picking them at random for each shot. This latter approach would have yielded IID data as required by Bowker's test, but would have been experimentally impractical. As a workaround, we subsample our data to mimic this latter approach, apply the Bowker test to the resulting subset of data, and repeat until the full data set has been tested.

More precisely, we initialized M to all zeros and used the same initial step as in Markov chain subsampling: for every circuit ℓ and every state k we effectively formed a list $L_{\ell k}$ of experimentally observed $|k\rangle \rightarrow |j\rangle$ transitions repeated by multiplicity. We then picked a circuit $\ell \sim \mathrm{uniform}(\{1, \ldots, N_{\mathrm{circ}}\})$ and an initial state $k \sim \mathrm{uniform}(\{0, \ldots, 2^n - 1\})$ uniformly at random, then drew a proposed jump $j \sim \mathrm{uniform}(L_{\ell,k})$. We removed an occurrence of j from $L_{\ell,k}$ and increased m_{jk} by 1. We repeated this process until we hit an empty $L_{\ell,k}$. Unlike in Markov chain subsampling, we picked k uniformly at random in each step, rather than accepting/rejecting the previous j to form a Markov chain. Since this process produces a random subset of IID data, we call it *IID subsampling*. We then applied Bowker's test to the resulting M and computed p_{sym}. Finally, we repeated these steps many times (starting over with the full data set each time) until every datum had been used in at least one Bowker test with overwhelming probability. In effect, we tested for symmetry in the full data set by testing for it in many random subsets which, together, covered the full set. There is some ambiguity in how to combine the p-values resulting from the different random subsets, especially since they are correlated (so for a fixed data set they need not be uniformly distributed, even under the null hypothesis). However, the observed p-values, given in Section IV C, were so large that they clearly did not show any statistically significant asymmetry, regardless of how they are combined.

Note that exact $Q(\boldsymbol{s'}|\,\boldsymbol{s}) = Q(\boldsymbol{s}\,|\boldsymbol{s'})$ symmetry is impossible in a complex engineered system like a quantum computer, or a classical computer for that matter. Given enough data, it will always be possible to resolve some slight asymmetry. However, this is less of a practical issue and more of a philosophical one. Practically, if we assume the ratio $Q(\boldsymbol{s}\,|\boldsymbol{s'})/Q(\boldsymbol{s'}|\,\boldsymbol{s}) = 1$ in the M-H acceptance probability (Eq. (3) of the main text) but the true value is slightly different, the resulting Markov chain may converge to a slightly different distribution than intended [41], which may slightly bias thermal average estimates like in Eq. (10) of the main text. We could not resolve any such biases in our experiments. More broadly, we propose the following rule of thumb: if the (potential) asymmetry in Q is too small to detect from the data at hand, it is probably too small to make much difference in your MCMC results. This heuristic cuts both ways of course, and suggests that long Markov chains are more sensitive to such asymmetry. We raise this issue to clarify the following point: we do not claim that any of our experiments realized a $\boldsymbol{Q}$ that was *exactly* symmetric, only that it was symmetric enough for practical purposes.

Finally, there are two main ways to handle empty pairs of bins ($m_{jk} = m_{kj} = 0$), to our knowledge. In both, the corresponding $0/0$ terms in Eq. (S23) are set to zero, since they give no evidence of asymmetry. (A different way to motivate this is to substitute m_{jk} with its expectation value $N\,\Pr[(j,k)]$ in Eq. (S23), and likewise for m_{kj}. The corresponding term in χ^2 tends towards zero as N decreases.) The traditional Bowker test uses $k = \binom{2^n}{2}$ chi-squared degrees of freedom regardless of the observed data; however, a variant (used in Stata statistics software [42], for instance) accounts for empty bin pairs by using $k = \binom{2^n}{2} - N_{\mathrm{empty}}$ instead, where N_{empty} is the number of empty pairs. We call this variant the modified Bowker test. We give p-values for both variants in Section IV C; when there are empty bin pairs the traditional Bowker test tends to be more conservative (i.e., to give larger p-values) than the modified test.

3. Bootstrapping

In Fig. 3 of the main text and similar figures in Section IV C, we computed δ confidence intervals (CIs) for our experiments through bootstrapping. However, we faced a similar issue as in Section IV B 2, in that bootstrapping assumes IID data, whereas our experiments naturally produced correlated data. We therefore employed IID subsampling to compute CIs. That is, for every T we generated an IID subsample M of the full dataset as described Section IV B 2. We then used the `stats.bootstrap` function from the SciPy library [19] with the “basic” (i.e., reverse percentile) setting to resample this M (rather than the full dataset) 200 times to estimate the 99% confidence interval for δ at this T. (We picked the most stringent bootstrap settings possible within a reasonable computational cost. The 99% confidence level was chosen to match the 1% significant threshold used in Section IV B 2.) We estimated $Q(j|k)$ from M using

$$\hat{Q}(j|k) = \frac{m_{jk}}{\sum_{j=0}^{2^n-1} m_{jk}}, \tag{S25}$$

then estimated $\boldsymbol{P}$ and δ following Eq. (S21), and used this latter estimator for bootstrapping. We repeated this process for every T, generating a new subsample M for each temperature. This means that while we used the full dataset to form the point estimate of δ at each T, i.e., the solid line in Fig. 3 and the like, we used different random subsets of the data to compute the CIs at each T. This subsampling, together with the stochastic nature of bootstrapping, is why the estimated δ forms a smooth curve while the CIs fluctuate more noticeably. This approach also means that the computed CIs are almost certainly pessimistic (i.e., overly broad), since they do not use all the data. Finally, note that the CIs at adjacent temperatures all agree reasonably well (in that there are no large, erratic fluctuations), despite using different random subsets of data. This self-consistency lends credence to the bootstrap settings described above, and to this approach to estimating δ CIs more broadly.

Note that the numerical data used to estimate δ for the S-W and Wolff cluster algorithms in Figs. S16, S21 and S28 was IID by design, so we could bootstrap it directly, with no need to subsample it first.

C. Supplemental data

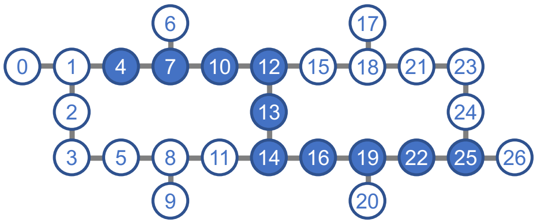
In this section we give some supplementary figures of merit for the $n = 10$ experiment presented in the main text, as well as data for two other experiments on illustrative model instances with $n = 8$ and $n = 9$ spins.

1. $n = 10$ *instance*

The qubits used for the $n = 10$ experiment presented in the main text, together with representative figures of merit from the last calibration before the experiment, are shown in Fig. S14. The problem instance is defined by the couplings and fields

$$\begin{aligned}(J_{j+1,j})_{j=1}^{n-1} &= \big(-0.99121054,\ 0.84436089,\ -0.83043895,\ 0.95766024,\ -1.02814718,\ 1.24969204,\\ &\qquad 0.81649925,\ -0.92147578,\ 1.11394418\big)\\ (h_j)_{j=1}^{n} &= \big(0.47921821,\ 0.10207621,\ -0.4780673,\ -0.39407213,\ 0.15239487,\ 0.44938277,\\ &\qquad 0.91715616,\ 0.73303354,\ 0.40145444,\ 0.55915183\big),\end{aligned} \tag{S26}$$

with all other J_{jk} equal to zero.



CNOT error (mean)	0.66%
CNOT error (worst)	0.98%
Readout assignment error (mean)	2.06%
Readout assignment error (worst)	4.94%

FIG. S14. **Device information for $n = 10$ experiment**. Left: the 1D chain of qubits on *ibmq_mumbai* used for this experiment, where Ising model spins $1, \ldots, 10$ were mapped to qubits $4, \ldots, 25$ respectively. Right: representative figures of merit for this chain of qubits at the time of the experiment.

Let $\boldsymbol{Q}_{\text{th}}$ be the $2^n \times 2^n$ matrix of proposal probabilities for our algorithm obtained through numerical simulation, and let $\hat{\boldsymbol{Q}}_{\text{exp}}$ be that of experimentally estimated probabilities defined by Eq. (S20). The total variation (TV) distance, defined as

$$\|\vec{p} - \vec{q}\|_{\text{TV}} = \frac{1}{2} \sum_{i=0}^{2^n-1} \left|p_i - q_i\right| = \frac{1}{2}\|\vec{p} - \vec{q}\|_1 \tag{S27}$$

for distributions $\vec{p}, \vec{q} \in \mathbb{R}^{2^n}$, is the most common measure of distance between distributions in MCMC. Accordingly, we use

$$\text{TV error} = \left\|\boldsymbol{Q}_{\text{th}} - \hat{\boldsymbol{Q}}_{\text{exp}}\right\|_{\text{TV,avg}} = \frac{1}{2^{n+1}} \sum_{i,j=0}^{2^n-1} \left|(\boldsymbol{Q}_{\text{th}})_{ij} - (\hat{\boldsymbol{Q}}_{\text{exp}})_{ij}\right| \tag{S28}$$

as our measure of experimental error; namely, the average TV distance between all columns of $\boldsymbol{Q}_{\text{th}}$ and $\hat{\boldsymbol{Q}}_{\text{exp}}$ (each of which is a probability distribution). The extreme values of 0 and 1 represent the smallest and largest possible error, respectively. Since our algorithm only uses initial states and measurements in the computational basis, the experimental error is completely captured by comparing the quantum transition probabilities in $\boldsymbol{Q}_{\text{th}}$ and $\hat{\boldsymbol{Q}}_{\text{exp}}$. This makes TV error a better figure of merit in this context than more typical measures of distance between quantum channels (e.g., process fidelity, diamond distance etc.) encompassing all possible initial states and measurement bases [43], most of which are *a priori* irrelevant here. The TV error for this experiment, together with other experimental parameters defined in Section IV A 3, is given in the left panel of Fig. S15. For comparison, the average TV distances between $\hat{\boldsymbol{Q}}_{\text{exp}}$ and the uniform proposal, the local proposal, and the identity matrix I are larger: 0.259, 0.953 and 0.987 respectively.

Fig. S15 also gives p-values from the traditional and modified Bowker tests, averaged over 200 IID subsamples as described in Section IV B 2. Both are well above the 1% significance threshold, and are therefore consistent with the experimental quantum proposal mechanism being symmetric. Typically, about 15% of the full dataset makes it into each IID subsample used for the Bowker tests. There were a total of 5.76×10^7 shots, so the expected number of data points that did not make it into any subsample is approximately $5.76 \times 10^7 \times (1 - 0.15)^{200} \sim 10^{-7} \ll 1$. The

N_{twirl}	6
N_{shots}	4×10^4
Total shots	5.76×10^7
TV error	0.206
$\langle p_{\text{sym}} \rangle$ traditional	36.7%
$\langle p_{\text{sym}} \rangle$ modified	25.1%

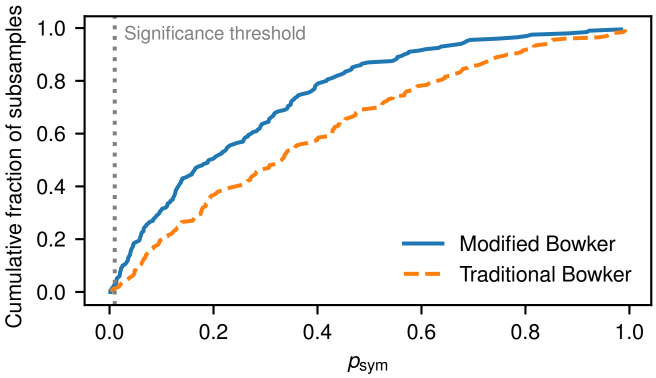


FIG. S15. **Parameters and figures of merit for $n = 10$ experiment.** Left: experimental parameters (top) and the resulting figures of merit (bottom). Right: Cumulative histograms showing the distribution of p-values from Bowker tests run on random subsets of the full dataset. The 1% significance threshold used throughout is shown for comparison.

200 subsamples used to test for symmetry therefore cover the full dataset. The integrated histogram of p-values for both Bowker test variants over these random subsamples are shown explicitly in the right panel of Fig. S15. Neither is sharply concentrated on $p_{\text{sym}} \lesssim 1\%$, which constitutes further evidence of symmetry.

Figs. S16, S17 and S18 show the same experimental data as Figs. 3 and 5a-b of the main text, but in different ways. Specifically, Fig. S16 shows the same results as Fig. 3, but also plots the spectral gaps for the five MCMC cluster algorithms introduced in Sec III D. As discussed in the main text, they are substantially more complicated than the other classical MCMC algorithms considered, but offer no significant δ improvement in the low-T regime of interest, which is why they are relegated to here. For completeness, Fig. S17 similarly shows the absolute spectral for two M-H variants discussed in Sections III C 3 and III C 4: lazy M-H and the Gibbs sampler acceptance probability. The quantum enhancement in δ at low T is nearly identical in both variants. Finally, Fig. S18 shows the same proposal probabilities $Q(\boldsymbol{s'} | \boldsymbol{s})$ as Fig. 5a-b, but with the spin configurations sorted lexicographically, rather than by increasing E.

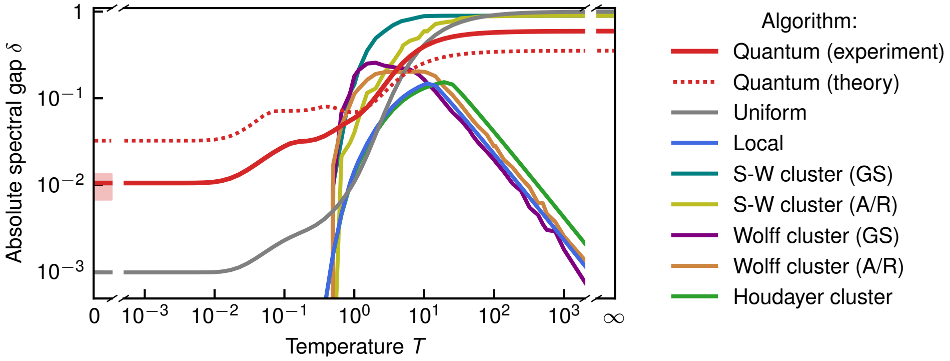
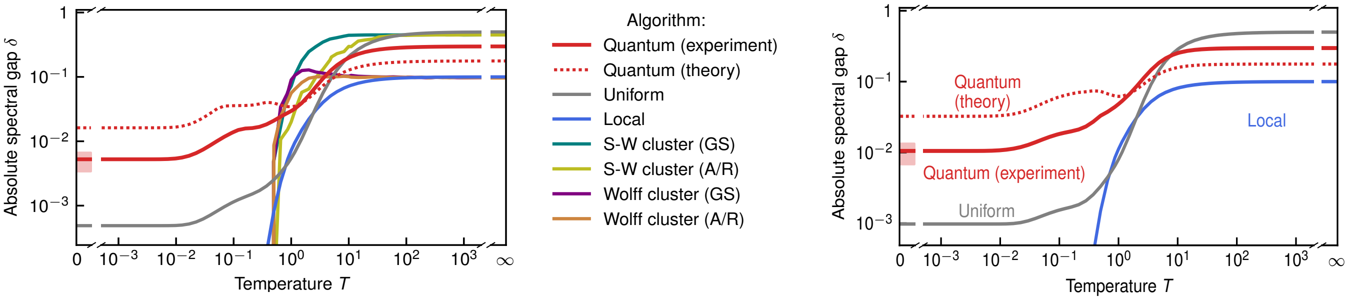


FIG. S16. **Convergence rates for $n = 10$ instance.** The absolute spectral gap δ for various MCMC algorithms on the same $n = 10$ model instance. The quantum, local and uniform proposal strategies were combined with the Metropolis-Hastings acceptance probability (3) of the main text. The data shown for these is the same as in Fig. 3 of the main text. It is re-plotted here for comparison with the five cluster algorithms discussed in Section III D. "GS" and "A/R" denote "ghost spin" and "accept/reject" versions respectively of the Swendsen-Wang (S-W) and Wolff algorithms. Error bands, where present, denote 99% confidence intervals found through basic (i.e., reverse percentile) bootstrapping using 200 resamples independently for each T. For the experimental realization of our quantum algorithm we bootstrapped random subsets of the full dataset, which comprises 5.76×10^7 shots, as described in Sec. IV B 3. For the S-W and Wolff algorithms there was no need for such subsampling. Rather, we formed point estimates and confidence intervals for their absolute spectral gaps at different temperatures by generating 10^5 IID cluster moves separately for each T.



(a) Lazy M-H

(b) Gibbs sampler

FIG. S17. **Convergence rates for $n = 10$ instance—variants.** The absolute spectral gap δ for various MCMC algorithms on the same $n = 10$ model instance. (a) Lazy versions of the Markov chains considered in Fig. 3 of the main text and Fig. S16. The Houdayer cluster algorithm is not shown since there is some ambiguity in how a lazy version should be defined, and because its δ at low T is set by a positive eigenvalue, meaning lazy variants should offer no speedup anyways. “GS” and “A/R” denote “ghost spin” and “accept/reject” versions respectively of the Swendsen-Wang (S-W) and Wolff algorithms. (b) Absolute spectral gaps based on the Gibbs sampler acceptance probability in Eq. (S8), rather than the M-H probability in Eq. (3) of the main text. Error bands in both panels, where present, denote 99% bootstrap confidence intervals constructed identically to those in Fig. S16.

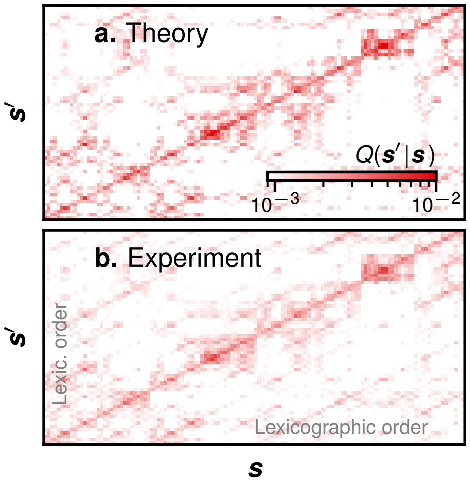


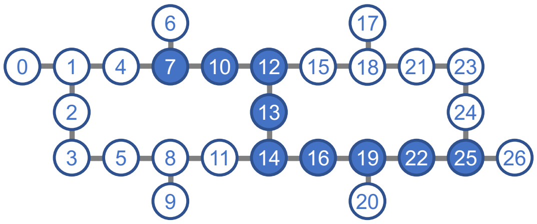
FIG. S18. The same data shown in Figs. 5a-b of the main text, but with $\boldsymbol{s}$ and $\boldsymbol{s'}$ sorted in lexicographic order, rather than by increasing Ising energy E.

2. $n = 9$ *instance*

We implemented our algorithm for another illustrative model instance with $n = 9$ spins. We present the experimental results here, which are analogous to those for the $n = 10$ instance from the main text. This $n = 9$ instance is defined by the couplings and fields

$$\begin{aligned}(J_{j+1,j})_{j=1}^{n-1} &= \big(0.89504271,\ 0.85382636,\ 1.01494031,\ 1.04243257,\ 1.19556769,\ 1.03985729\\ &\qquad\qquad 1.15258874,\ 1.17542484\big)\\ (h_j)_{j=1}^{n} &= \big(-0.41017914,\ 0.69312341,\ 0.61866226,\ 0.45567707,\ -0.32062001,\ 0.31684772\\ &\qquad\qquad -0.62064839,\ 0.04672033,\ -0.48091904\big),\end{aligned} \tag{S29}$$

with all other J_{jk} equal to zero. Its four lowest-E configurations are all local minima, and the lowest three are nearly degenerate: at $T = 0.1$ they have Boltzmann probabilities of approximately 37%, 36% and 27% for the ground, first- and second-excited configurations respectively. The qubits used in this experiment, together with representative figures of merit from the last calibration before the experiment, are shown in Fig. S19.



CNOT error (mean)	0.71%
CNOT error (worst)	0.88%
Readout assignment error (mean)	1.97%
Readout assignment error (worst)	4.53%

FIG. S19. **Device information for $n = 9$ experiment**. Left: the 1D chain of qubits on *ibmq_mumbai* used for this experiment, where Ising model spins $1, \ldots, 9$ were mapped to qubits $7, \ldots, 25$ respectively. Right: representative figures of merit for this chain of qubits at the time of the experiment.

The TV error for this experiment, together with other experimental parameters defined in Section IV A 3, is given in the left panel of Fig. S20. For comparison, the average TV distances between $\hat{\boldsymbol{Q}}_{\text{exp}}$ for this instance and the uniform proposal, the local proposal, and the identity matrix I are larger: 0.233, 0.935 and 0.980 respectively. Fig. S20 also gives p-values from the traditional and modified Bowker tests, averaged over 200 IID subsamples as described in Section IV B 2. Both are well above the 1% significance threshold, and are therefore consistent with the experimental quantum proposal mechanism being symmetric. Typically, about 30% of the full dataset makes it into each IID subsample used for the Bowker tests. There were a total of 2.88×10^7 shots, so the expected number of data points that did not make it into any subsample is approximately $2.88 \times 10^7 \times (1 - 0.3)^{200} \sim 10^{-24} \ll 1$. The 200 subsamples used to test for symmetry therefore cover the full dataset. The integrated histogram of p-values for both Bowker test variants over these random subsamples are shown explicitly in the right panel of Fig. S20. The two agree almost exactly since the data is denser in this experiment than in the $n = 10$ one (i.e., the ratio of total shots to the number of possible transitions is larger), so there are few empty pairs of bins. The distributions are more sharply concentrated around small p_{sym} than the corresponding $n = 10$ ones in Fig. S15, which we attribute to having used a smaller number of random twirls N_{twirl} per γ and t here. Nevertheless, their means are both well above the 1% significance threshold for asymmetry.

Figs. S21–S24 show results analogous to Figs. 3–5 from the main text for this experiment. Specifically, the inferred absolute spectral gap δ for M-H, as a function of temperature, is shown in Fig. S21. As in the $n = 10$ experiment, there is a significant quantum enhancement in δ at low T. For completeness, Fig. S22 similarly shows the absolute spectral gap for two M-H variants discussed in Sections III C 3 and III C 4: lazy M-H and the Gibbs sampler acceptance probability. The quantum enhancement in δ at low T is nearly identical in both variants. Fig. S23a shows the magnetization $m(\boldsymbol{s})$ for illustrative MCMC trajectories at $T = 0.1$. Our quantum algorithm jumps between the lowest-E configurations noticeably more often than the classical alternatives. Accordingly, the running average estimate for $\langle m \rangle_\mu$ converges more quickly to the true value of $\langle m \rangle_\mu \approx 0.35$ in Fig. S23b. Fig. S24 shows the distribution of jumps for our quantum algorithm, and reveals the same enhancement mechanism discussed in the main text for $n = 10$. Finally, Fig. S25 shows the same proposal probabilities $Q(\boldsymbol{s}'|\,\boldsymbol{s})$ as Figs. S24a-b, but with the spin configurations sorted lexicographically, rather than by increasing E.

N_{twirl}	4
N_{shots}	3×10^4
Total shots	2.88×10^7
TV error	0.189
$\langle p_{\mathrm{sym}} \rangle$ traditional	12.7%
$\langle p_{\mathrm{sym}} \rangle$ modified	12.7%

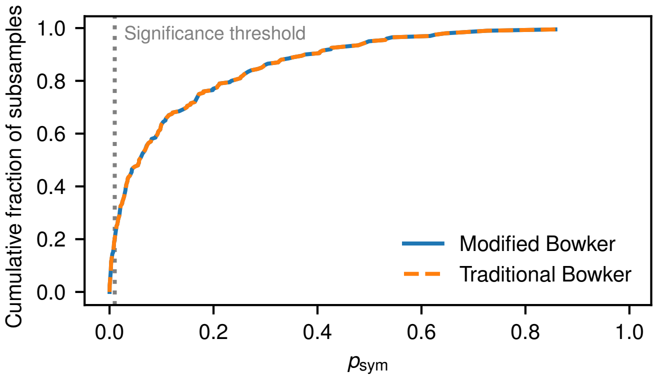


FIG. S20. **Parameters and figures of merit for $n = 9$ experiment.** Left: experimental parameters (top) and the resulting figures of merit (bottom). Right: Cumulative histograms showing the distribution of p-values from Bowker tests run on random subsets of the full dataset. The 1% significance threshold used throughout is shown for comparison.

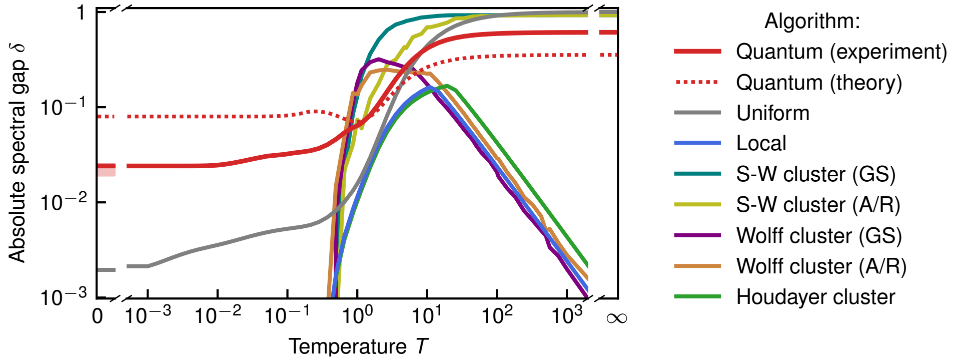
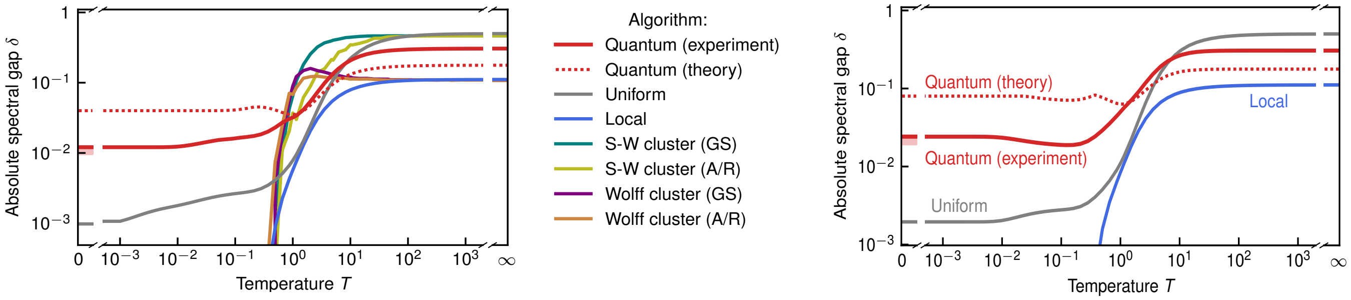


FIG. S21. **Convergence rates for $n = 9$ instance.** The absolute spectral gap δ for various MCMC algorithms on the same $n = 9$ model instance. This plot was constructed in the same way as Fig. S16. The quantum, local and uniform proposal strategies were combined with the Metropolis-Hastings acceptance probability (3) of the main text, and the cluster algorithms are discussed in Section III D. "GS" and "A/R" denote "ghost spin" and "accept/reject" versions respectively of the Swendsen-Wang (S-W) and Wolff algorithms. Error bands, where present, denote 99% confidence intervals found through basic (i.e., reverse percentile) bootstrapping using 200 resamples independently for each T. For the experimental realization of our quantum algorithm we bootstrapped random subsets of the full dataset, which comprises 2.88×10^7 shots, as described in Sec. IV B 3. For the S-W and Wolff algorithms there was no need for such subsampling. Rather, we formed point estimates and confidence intervals for their absolute spectral gaps at different temperatures by generating 10^5 IID cluster moves separately for each T.



(a) Lazy M-H

(b) Gibbs sampler

FIG. S22. **Convergence rates for $n = 9$ instance—variants.** The absolute spectral gap δ for various MCMC algorithms on the same $n = 9$ model instance. (a) Lazy versions of the Markov chains considered in Fig. S21. The Houdayer cluster algorithm is not shown since there is some ambiguity in how a lazy version should be defined, and because its δ at low T is set by a positive eigenvalue, meaning lazy variants should offer no speedup anyways. "GS" and "A/R" denote "ghost spin" and "accept/reject" versions respectively of the Swendsen-Wang (S-W) and Wolff algorithms. (b) Absolute spectral gaps based on the Gibbs sampler acceptance probability in Eq. (S8), rather than the M-H probability in Eq. (3) of the main text. Error bands in both panels, where present, denote 99% bootstrap confidence intervals constructed identically to those in Fig. S21.

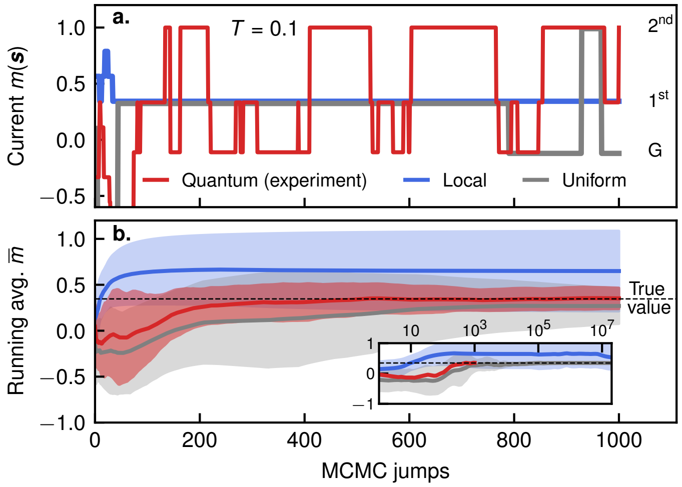


FIG. S23. **Magnetization estimate experiment with $n = 9$.** Analogous data to Fig. 4 of the main text but for the $n = 9$ instance, collected as described in Sec. IV B 1. **a.** The current magnetization $m(\boldsymbol{s}^{(j)})$ for individual Markov chains after j iterations. Each chain illustrates a different proposal strategy with uniformly random initialization. Arrows indicate the magnetization of the ground (G), 1st and 2nd excited configurations. **b.** Convergence of the running average $\bar{m}^{(j)} = \frac{1}{j}\sum_{k=0}^{j} m(\boldsymbol{s}^{(k)})$ from MCMC trajectories to the true value of $\langle m \rangle_\mu$ for different proposal strategies. For each strategy, the lines and error bands show the mean and standard deviation, respectively, of $\bar{m}^{(j)}$ over 10 independent chains. The inset depicts the same chains over more iterations. We do not use a burn-in period or thinning (i.e., the running average starts at $k = 0$ and includes every iteration up to $k = j$), as these practices would introduce hyperparameters that complicate the interpretation. Both panels are for $T = 0.1$ and use the Metropolis-Hastings acceptance probability in Eq. (3) of the main text.

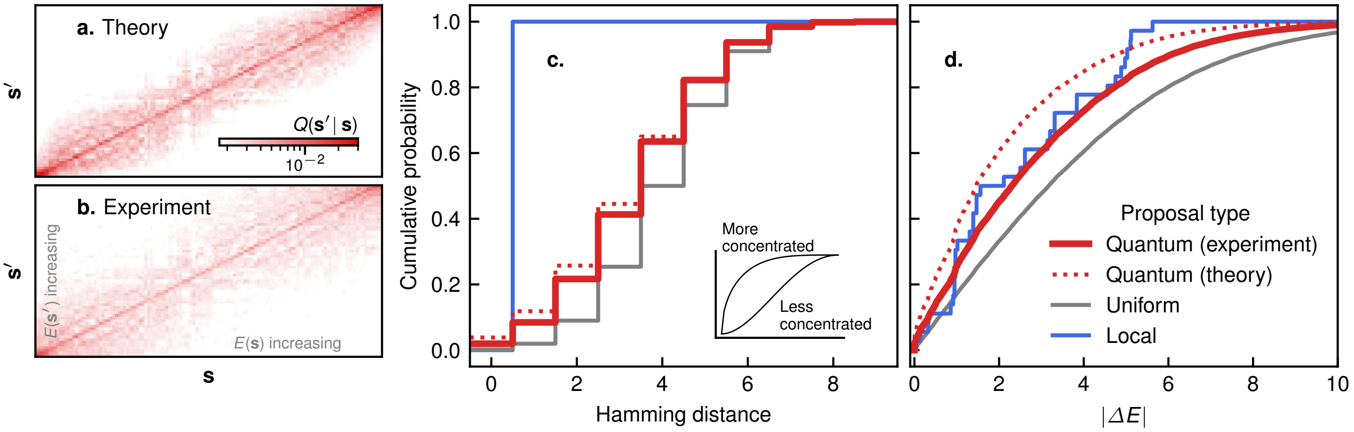


FIG. S24. **Quantum speedup mechanism for $n = 9$.** This figure shows data analogous to Fig. 5 of the main text, but for the $n = 9$ instance. **a.** The classically-simulated probabilities of $\boldsymbol{s} \to \boldsymbol{s}'$ proposals in our quantum algorithm, represented as a $2^n \times 2^n$ matrix whose columns are independent histograms. Both the initial and proposed configurations are sorted by increasing Ising energy E. The same data is shown with lexicographic ordering in Fig. S25. **b.** The estimated proposal probabilities for our algorithm's experimental realization. We estimated each $Q(\boldsymbol{s}' \mid \boldsymbol{s})$ as described in Sec. IV B. **c.** The probability distributions of Hamming distance between current ($\boldsymbol{s}$) and proposed ($\boldsymbol{s}'$) configurations, for a uniformly random current configuration. That of the experiment uses the estimated probabilities from panel b, while the rest were computed exactly. **d.** The analogous distributions for $|\Delta E| = |E(\boldsymbol{s}') - E(\boldsymbol{s})|$ of proposed jumps. Each distribution is depicted in full detail through its cumulative distribution function, with no binning. None of the panels depend on T or on the choice of acceptance probability.

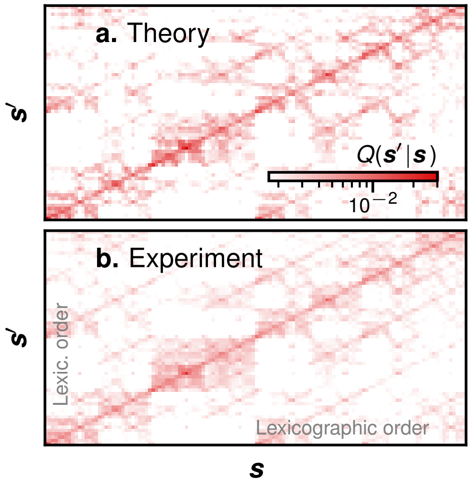


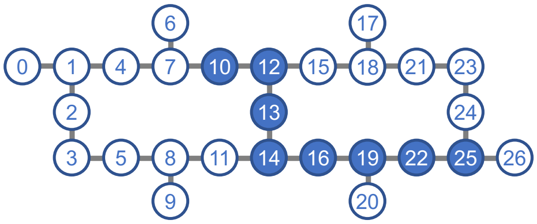
FIG. S25. The same data shown in Figs. S24a-b, but with $\boldsymbol{s}$ and $\boldsymbol{s'}$ sorted in lexicographic order, rather than by increasing Ising energy E.

3. $n = 8$ instance

We implemented our algorithm for another illustrative model instance with $n = 8$ spins. We present the experimental results here, which are analogous to those for the $n = 10$ instance from the main text. This $n = 8$ instance is defined by the couplings and fields

$$
\begin{aligned}
(J_{j+1,j})_{j=1}^{n-1} &= \big(-0.94496973,\ 1.01674697,\ 1.05072852,\ 1.07515862,\ 0.90289512,\ 0.98594583,\ 0.9361144\big) \qquad \text{(S30)}\\
(h_j)_{j=1}^{n} &= \big(0.53727449,\ -0.24671696,\ 0.4930372,\ -0.92341807,\ 0.75710839,\ -0.65808939,\ 0.30686208,\ 0.68948975\big),
\end{aligned}
$$

with all other J_{jk} equal to zero. Its four lowest-E configurations are all local minima, and the lowest three are nearly degenerate: at $T = 0.1$ they have Boltzmann probabilities of approximately 39%, 32% and 26% for the ground, first- and second-excited configurations respectively. The qubits used in this experiment, together with representative figures of merit from the last calibration before the experiment, are shown in Fig. S26.



CNOT error (mean)	0.70%
CNOT error (worst)	0.88%
Readout assignment error (mean)	2.05%
Readout assignment error (worst)	4.53%

FIG. S26. **Device information for $n = 8$ experiment**. Left: the 1D chain of qubits on *ibmq_mumbai* used for this experiment, where Ising model spins $1, \ldots, 8$ were mapped to qubits $10, \ldots, 25$ respectively. Right: representative figures of merit for this chain of qubits at the time of the experiment.

The TV error for this experiment, together with other experimental parameters defined in Section IV A 3, is given in the left panel of Fig. S27. For comparison, the average TV distances between $\hat{\boldsymbol{Q}}_{\text{exp}}$ for this instance and the uniform proposal, the local proposal, and the identity matrix I are larger: 0.238, 0.911 and 0.971 respectively. Fig. S27 also gives p-values from the traditional and modified Bowker tests, averaged over 200 IID subsamples as described in Section IV B 2. Both are well above the 1% significance threshold, and are therefore consistent with the experimental quantum proposal mechanism being symmetric. Typically, about 20% of the full dataset makes it into each IID subsample used for the Bowker tests. There were a total of 9.6×10^6 shots, so the expected number of data points that did not make it into any subsample is approximately $9.6 \times 10^6 \times (1 - 0.2)^{200} \sim 10^{-13} \ll 1$. The 200 subsamples used to test for symmetry therefore cover the full dataset. The integrated histogram of p-values for both Bowker test variants over these random subsamples are shown explicitly in the right panel of Fig. S27. The two agree almost exactly since the data is denser in this experiment than in the $n = 10$ one (i.e., the ratio of total shots to the number of possible transitions is larger), so there are few empty pairs of bins. Neither is sharply concentrated on $p_{\text{sym}} \lesssim 1\%$, which constitutes further evidence of symmetry.

N_{twirl}	4
N_{shots}	10^4
Total shots	9.6×10^6
TV error	0.177
$\langle p_{\text{sym}} \rangle$ traditional	41.4%
$\langle p_{\text{sym}} \rangle$ modified	41.4%

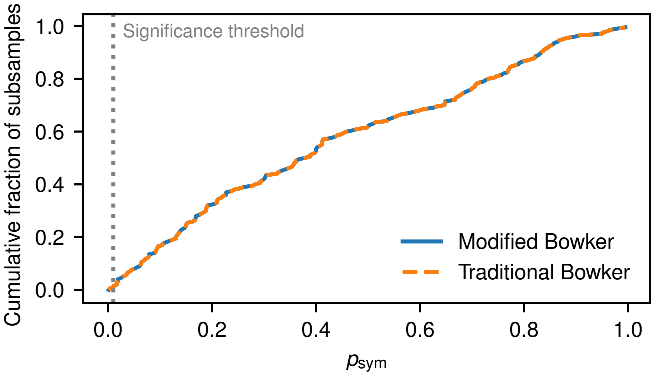


FIG. S27. **Parameters and figures of merit for $n = 8$ experiment.** Left: experimental parameters (top) and the resulting figures of merit (bottom). Right: Cumulative histograms showing the distribution of p-values from Bowker tests run on random subsets of the full dataset. The 1% significance threshold used throughout is shown for comparison.

Figs. S28–S32 show results analogous to Figs. 3–5 from the main text for this experiment. Specifically, the inferred absolute spectral gap δ for M-H, as a function of temperature, is shown in Fig. S28. As in the $n = 9$ and 10 experiments, there is a significant quantum enhancement in δ at low T. For completeness, Fig. S29 similarly shows the absolute spectral gap for two M-H variants discussed in Sections III C 3 and III C 4: lazy M-H and the Gibbs sampler acceptance probability. The quantum enhancement in δ at low T is nearly identical in both variants. Fig. S30a shows the magnetization $m(\boldsymbol{s})$ for illustrative MCMC trajectories at $T = 0.1$. Our quantum algorithm jumps between the lowest-E configurations noticeably more often than the classical alternatives. Accordingly, the running average estimate for $\langle m \rangle_\mu$ converges more quickly to the true value of $\langle m \rangle_\mu \approx -0.16$ in Fig. S30b. Fig. S31 shows the distribution of jumps for our quantum algorithm, and reveals the same enhancement mechanism discussed in the main text for $n = 10$. Finally, Fig. S32 shows the same proposal probabilities $Q(\boldsymbol{s}'|\,\boldsymbol{s})$ as Figs. S31a-b, but with the spin configurations sorted lexicographically, rather than by increasing E.

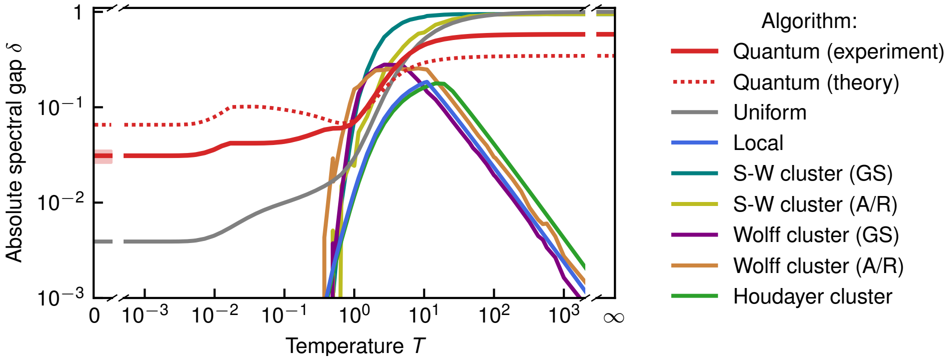
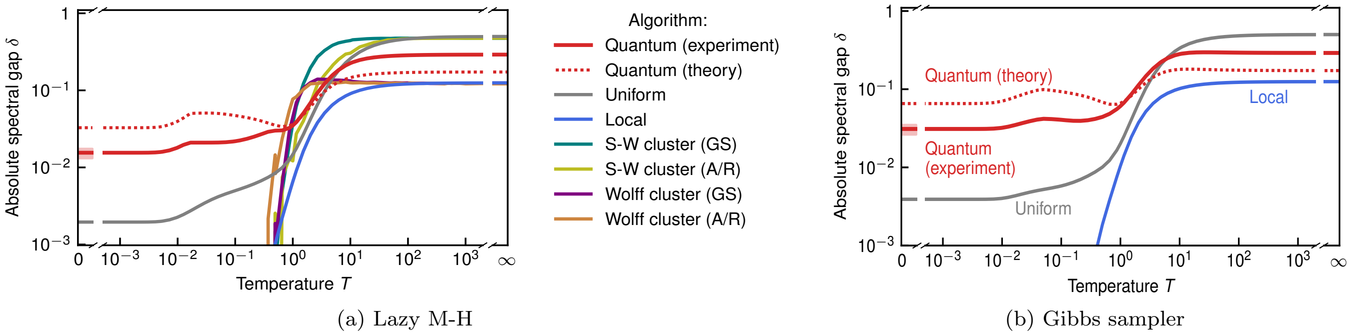


FIG. S28. **Convergence rates for $n = 8$ instance.** The absolute spectral gap δ for various MCMC algorithms on the same $n = 8$ model instance. This plot was constructed in the same way as Fig. S16. The quantum, local and uniform proposal strategies were combined with the Metropolis-Hastings acceptance probability (3) of the main text, and the cluster algorithms are discussed in Section III D. "GS" and "A/R" denote "ghost spin" and "accept/reject" versions respectively of the Swendsen-Wang (S-W) and Wolff algorithms. Error bands, where present, denote 99% confidence intervals found through basic (i.e., reverse percentile) bootstrapping using 200 resamples independently for each T. For the experimental realization of our quantum algorithm we bootstrapped random subsets of the full dataset, which comprises 9.6×10^6 shots, as described in Sec. IV B 3. For the S-W and Wolff algorithms there was no need for such subsampling. Rather, we formed point estimates and confidence intervals for their absolute spectral gaps at different temperatures by generating 10^5 IID cluster moves separately for each T.



(a) Lazy M-H

(b) Gibbs sampler

FIG. S29. **Convergence rates for $n = 8$ instance—variants.** The absolute spectral gap δ for various MCMC algorithms on the same $n = 8$ model instance. (a) Lazy versions of the Markov chains considered in Fig. S28. The Houdayer cluster algorithm is not shown since there is some ambiguity in how a lazy version should be defined, and because its δ at low T is set by a positive eigenvalue, meaning lazy variants should offer no speedup anyways. "GS" and "A/R" denote "ghost spin" and "accept/reject" versions respectively of the Swendsen-Wang (S-W) and Wolff algorithms. (b) Absolute spectral gaps based on the Gibbs sampler acceptance probability in Eq. (S8), rather than the M-H probability in Eq. (3) of the main text. Error bands in both panels, where present, denote 99% bootstrap confidence intervals constructed identically to those in Fig. S28.

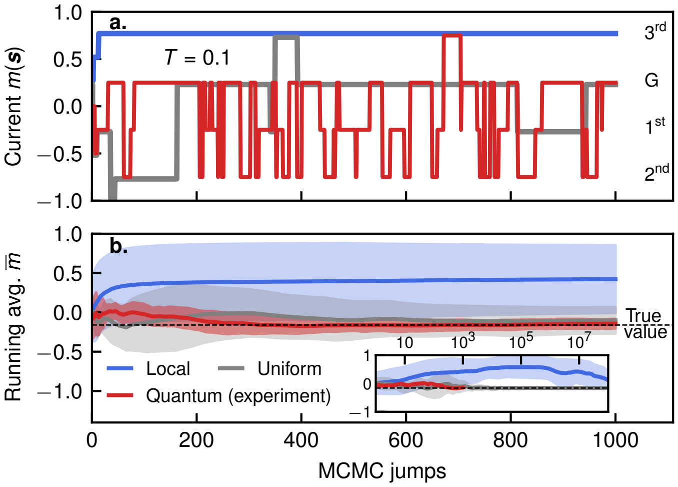


FIG. S30. **Magnetization estimate experiment with $n = 8$.** Analogous data to Fig. 4 of the main text but for the $n = 8$ instance, collected as described in Sec. IV B 1. **a.** The current magnetization $m(\boldsymbol{s}^{(j)})$ for individual Markov chains after j iterations. Each chain illustrates a different proposal strategy with uniformly random initialization. Arrows indicate the magnetization of the ground (G), 1st, 2nd and 3rd excited configurations. **b.** Convergence of the running average $\bar{m}^{(j)} = \frac{1}{j}\sum_{k=0}^{j} m(\boldsymbol{s}^{(k)})$ from MCMC trajectories to the true value of $\langle m \rangle_\mu$ for different proposal strategies. For each strategy, the lines and error bands show the mean and standard deviation, respectively, of $\bar{m}^{(j)}$ over 10 independent chains. The inset depicts the same chains over more iterations. We do not use a burn-in period or thinning (i.e., the running average starts at $k = 0$ and includes every iteration up to $k = j$), as these practices would introduce hyperparameters that complicate the interpretation. Both panels are for $T = 0.1$ and use the Metropolis-Hastings acceptance probability in Eq. (3) of the main text.

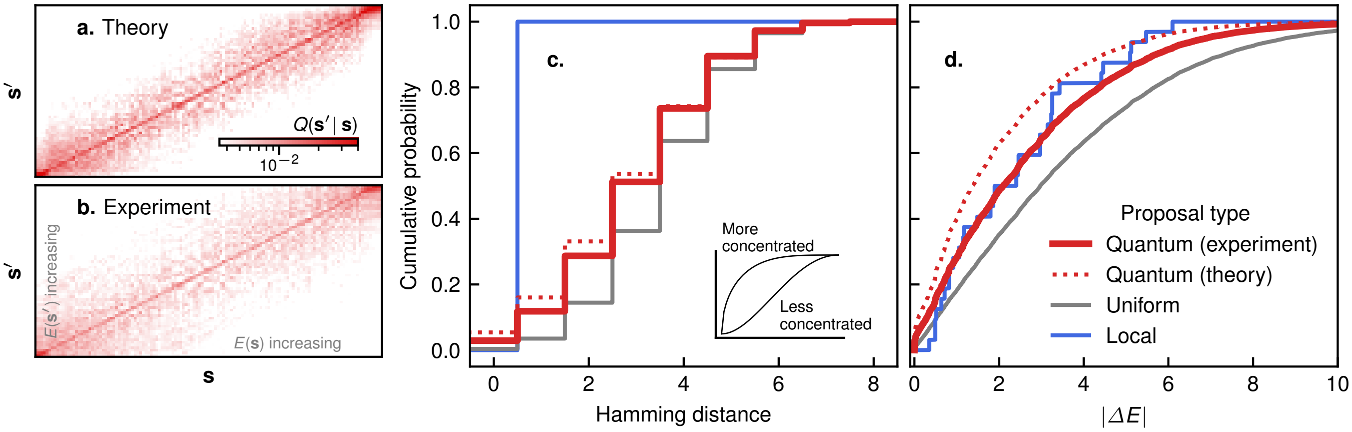


FIG. S31. **Quantum speedup mechanism for $n = 8$.** This figure shows data analogous to Fig. 5 of the main text, but for the $n = 8$ instance. **a.** The classically-simulated probabilities of $\boldsymbol{s} \to \boldsymbol{s'}$ proposals in our quantum algorithm, represented as a $2^n \times 2^n$ matrix whose columns are independent histograms. Both the initial and proposed configurations are sorted by increasing Ising energy E. The same data is shown with lexicographic ordering in Fig. S32. **b.** The estimated proposal probabilities for our algorithm's experimental realization. We estimated each $Q(\boldsymbol{s'} | \boldsymbol{s})$ as described in Sec. IV B. **c.** The probability distributions of Hamming distance between current ($\boldsymbol{s}$) and proposed ($\boldsymbol{s'}$) configurations, for a uniformly random current configuration. That of the experiment uses the estimated probabilities from panel b, while the rest were computed exactly. **d.** The analogous distributions for $|\Delta E| = |E(\boldsymbol{s'}) - E(\boldsymbol{s})|$ of proposed jumps. Each distribution is depicted in full detail through its cumulative distribution function, with no binning. None of the panels depend on T or on the choice of acceptance probability.

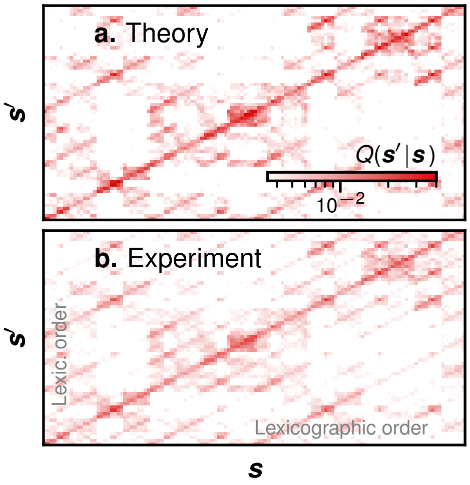


FIG. S32. The same data shown in Figs. S31a-b, but with $\boldsymbol{s}$ and $\boldsymbol{s'}$ sorted in lexicographic order, rather than by increasing Ising energy E.

V. QUANTUM PROPOSAL STRATEGIES: MOTIVATION AND ALTERNATIVES

In this section we discuss the motivation for our quantum proposal strategy as well as possible alternatives.

A. Perturbative regime

We begin with the pertubative calculation to which the main text refers. Consider H as given by Eqs. (5) and (6) of the main text, where H_{mix} is some as-yet unspecified Hamiltonian with $\langle k|H_{\text{mix}}|j\rangle \in \mathbb{R}$ for all computational basis states $|j\rangle$ and $|k\rangle$. To emphasize the dependence on γ we will sometimes write $H = H(\gamma)$. We will compute the $|j\rangle \to |k\rangle$ quantum transition probability perturbatively to leading order in γ, assuming for now that γ is sufficiently small for perturbation theory to hold.

It is convenient to move to the interaction picture defined by αH_{prob}, where the Hamiltonian becomes

$$\tilde{H}(\gamma, t) = e^{it\alpha H_{\text{prob}}}\big[H(\gamma) - \alpha H_{\text{prob}}\big]e^{-it\alpha H_{\text{prob}}} = \gamma\left(e^{it\alpha H_{\text{prob}}} H_{\text{mix}}\, e^{-it\alpha H_{\text{prob}}} - \alpha H_{\text{prob}}\right). \tag{S31}$$

Let $U(t) = e^{-iH(\gamma)t}$ be the propagator from time 0 to t in the Schrödinger picture, and $\tilde{U}(t)$ be that in the interaction picture (satisfying $i\tilde{U}' = \tilde{H}\tilde{U}$ and $\tilde{U}(0) = I$). Since H_{prob} is diagonal in the computational basis, the quantum transition probabilities are simply the absolute-squared matrix elements of the propagator in either picture:

$$\Pr\Big(|j\rangle \to |k\rangle\Big) = \big|\langle k|\tilde{U}(t)|j\rangle\big|^2 = \big|\langle k|U(t)|j\rangle\big|^2. \tag{S32}$$

We can expand $\tilde{U}(t)$ perturbatively in γ through a Dyson series [44]:

$$\tilde{U}(t) = I - i\int_0^t \tilde{H}(\gamma, t')\, dt' + O(\gamma^2). \tag{S33}$$

To leading order in γ, its matrix elements are

$$\langle k|\tilde{U}(t)|j\rangle = \begin{cases} \gamma \int_0^t e^{it'\alpha\Delta E} dt' \langle k|H_{\text{mix}}|j\rangle + O(\gamma^2) & j \neq k \\ 1 - O(\gamma) & j = k, \end{cases} \tag{S34}$$

where $\Delta E = E(k) - E(j)$ is the difference in classical energies, defined in Eq. (1) of the main text, between spin configurations j and k. Using Eq. (S32) we find

$$\Pr\Big(|j\rangle \to |k\rangle\Big) = \begin{cases} \gamma^2 t^2 \operatorname{sinc}^2\left(\frac{t\alpha\Delta E}{2}\right)\big|\langle k|H_{\text{mix}}|j\rangle\big|^2 + O(\gamma^3) & j \neq k \\ 1 - O(\gamma^2) & j = k. \end{cases} \tag{S35}$$

The sinc-squared factor above acts as a bandpass filter for ΔE, since $\operatorname{sinc}^2(x)$ is of order unity within the passband $x \in [-\pi, \pi]$ but nearly 0 otherwise. Suppose that H_{mix} is dense, e.g., $\big|\langle k|H_{\text{mix}}|j\rangle\big| = 1$ as invoked in early formulations of Grover's algorithm using quantum walks [45]. Then writing $t = 2\pi/\alpha\epsilon$ for some energy scale ϵ, we see that nontrivial $|j\rangle \to |k\rangle$ quantum transitions occur predominantly when $|\Delta E| \lesssim \epsilon$. In other words, they occur primarily between configurations j and k whose classical energies $E(j) \approx E(k)$ are close on the scale set by ϵ, even if they are far in Hamming distance.

The same conclusion can be reached using a Magnus series rather than a Dyson series [46]. That is, we can define an effective Hamiltonian $\tilde{H}_{\text{eff}}$ term-by-term in powers of γ such that $\tilde{U}(t) = \exp(-it\tilde{H}_{\text{eff}})$ (with no time ordering). To leading order in γ, it has the form $\tilde{H}_{\text{eff}} = \frac{1}{t}\int_0^t \tilde{H}(\gamma, t')\, dt' + O(\gamma^2)$ and matrix elements

$$\langle k|\tilde{H}_{\text{eff}}|j\rangle = \frac{\gamma}{t}\int_0^t e^{it'\alpha\Delta E} dt' \langle k|H_{\text{mix}}|j\rangle + O(\gamma^2) \tag{S36}$$

in the computational basis. Invoking the rotating wave approximation, the fast-rotating elements (with $\Delta E \gtrsim 2\pi/\alpha t$) average out and nearly vanish, but the slow ones do not [47]. The effective Hamiltonian therefore primarily couples states $|j\rangle$ and $|k\rangle$ for which $E(j) \approx E(k)$, even if j and k are far in Hamming distance, causing $|j\rangle \to |k\rangle$ transitions between such states to dominate.

Both of these methods assume t to be sufficiently small; however, the same phenomenon also occurs at long times. We return to the Schrödinger picture for this calculation. Suppose one prepares a state $|j\rangle$, evolves it under $H(\gamma)$ for a time t and measures in the computational basis. For a random $t \sim \text{uniform}([0, t'])$, the probability of measuring $|k\rangle$ approaches

$$\Pr\Big(|j\rangle \to |k\rangle\Big) = \sum_{\ell=0}^{2^n-1} \big|\langle k|\lambda_\ell\rangle\langle\lambda_\ell|j\rangle\big|^2 \tag{S37}$$

as $t' \to \infty$ [48], where $\{|\lambda_\ell\rangle\}_{\ell=0}^{2^n-1}$ are the eigenvectors of $H(\gamma)$, indexed such that $\lim_{\gamma\to 0} |\lambda_j\rangle = |j\rangle$. For simplicity, we assume that $H(\gamma)$ has a non-degenerate spectrum. Expanding $|\lambda_\ell\rangle$ in powers of γ through time-independent, non-degenerate perturbation theory gives

$$\langle k|\lambda_\ell\rangle = \begin{cases} \frac{\gamma}{\alpha}\frac{\langle k|(H_{\text{mix}}-\alpha H_{\text{prob}})|\ell\rangle}{E_\ell - E_k} + O(\gamma^2) & k \neq \ell \\ 1 - O(\gamma) & k = \ell. \end{cases} \tag{S38}$$

Combining this with Eq. (S37), we get

$$\Pr\Big(|j\rangle \to |k\rangle\Big) = \begin{cases} 2\left(\frac{\gamma}{\alpha\Delta E}\right)^2 \big|\langle k|H_{\text{mix}}|j\rangle\big|^2 + O(\gamma^3) & j \neq k \\ 1 - O(\gamma^2) & j = k. \end{cases} \tag{S39}$$

This equation exhibits the same ΔE^{-2} scaling as the sinc-squared term of Eq. (S35), leading to the same conclusion: non-trivial transitions occur primarily between states with similar classical energies, even if they are far in Hamming distance. This last calculation, however, highlights the role of H's eigenvectors in driving quantum transitions between such states. Namely, these transitions occur because H has eigenvectors (specifically, $|\lambda_j\rangle$ and $|\lambda_k\rangle$) that overlap substantially with both $|j\rangle$ and $|k\rangle$ when $E(j) \approx E(k)$, but that are nearly orthogonal to other computational states $|\ell\rangle$ for which $E(j) \not\approx E(\ell)$.

A similar, albeit much more complicated, effect can also occur outside the perturbative regime for γ. Certain quantum spin glass Hamiltonians have been shown to possess eigenvectors with components concentrated on (potentially distant) low-E configurations forming local minima of similar energies [49–51]. These are the spin configurations that often cause bottlenecks in MCMC at low temperatures T, as discussed in the main text. Evolution under such Hamiltonians can therefore produce relatively frequent quantum transitions between these configurations, analogous to those in the previous, perturbative calculation. There are important advantages to using a large γ beyond the scope of perturbation theory, however: (i) H_{mix} can be sparse, which is easier to realize experimentally. That is, quantum transitions can occur with high probability between states $|j\rangle$ and $|k\rangle$ even if $\langle k|H_{\text{mix}}|j\rangle = 0$. (ii) Trivial $|j\rangle \to |j\rangle$ "transitions" need not dominate, as they do in the perturbative analysis above. (iii) The resulting dynamics can be hard to simulate classically. However, it is not clear *a priori* how the results of Refs. [49–51] translate into MCMC performance. For instance, the spin glasses studied in these works are idealizations of those we consider. We therefore view the perturbative calculations above, and Refs. [49–52], as providing motivation for our algorithm—which we then evaluate empirically—rather than any formal guarantee of performance.

B. Quantum phase estimation

The final calculation of the Section V A involved time evolution e^{-iHt} for a uniformly random t. This dephases an initial state $|j\rangle$ in the eigenbasis of H. One could therefore realize the same effect by preparing an initial state $|j\rangle$, measuring it in the eigenbasis of H, then measuring the resulting state in the computational basis. While it is not typically feasible to measure H directly, the conventional way to approximate such a measurement is through quantum phase estimation (QPE), as shown in Fig. S33. The measurement result from the top register encodes a k-bit approximation to an eigenvalue λ_ℓ of H, while the bottom register—just prior to measurement—is approximately in the corresponding eigenstate $|\lambda_\ell\rangle$. The measurement result $\boldsymbol{s'}$ from the bottom register gives the proposed move. It is easy to show that this realizes a symmetric proposal $Q(\boldsymbol{s'}|\boldsymbol{s}) = Q(\boldsymbol{s}|\boldsymbol{s'})$.

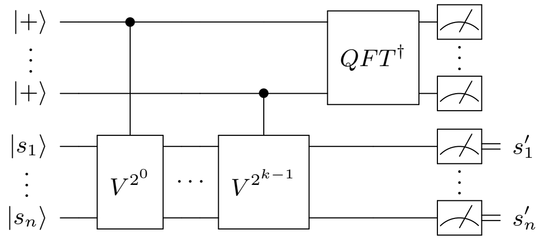


FIG. S33. An alternative proposal mechanism based on quantum phase estimation. Here $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $V = e^{-iH\tau}$ for some desired timescale τ, and QFT is the quantum Fourier transform. The circuit comprises two quantum registers: the top one consists of k qubits initialized in $|+\rangle^{\otimes k}$, while the bottom one consists of n qubits initialized in the computational state $|\boldsymbol{s}\rangle$.

While using QPE to propose moves as in Fig. S33 may be of theoretical interest, we see little practical appeal in this approach at the moment. The issue is that the measurement result from the top register is not currently used. This means that one could forego the quantum Fourier transform without affecting the bottom register. In fact, there is no reason at present to use the top register at all. Instead, one could achieve the same effect on the bottom register by picking a random $m \sim \text{uniform}(\{2^0, 2^1, \ldots, 2^{k-1}\})$, applying V^m (not conditioned on any quantum state) to $|\boldsymbol{s}\rangle$, then measuring in the computational basis. This latter scheme, in turn, is equivalent to propagating $|\boldsymbol{s}\rangle$ by e^{-iHt} for a random $t \sim \text{uniform}(\{\tau 2^0, \tau 2^1, \ldots, \tau 2^{k-1}\})$. In effect then, the only difference between this QPE method and the random-t method in Algorithm S1 is the set of possible values from which t is drawn.

C. Reverse annealing

We have focused so far on time-independent Hamiltonians H given by Eqs. (5) and (6) of the main text, where $\langle k|H_{\text{mix}}|j\rangle \in \mathbb{R}$ for all computational states $|j\rangle$ and $|k\rangle$. This ensures that H is real and symmetric (not just Hermitian), so it has real eigenvalues $\{\lambda_\ell\}_{\ell=0}^{2^n-1} \subset \mathbb{R}$ and a set of real orthonormal eigenvectors $\{|\lambda_\ell\rangle\}_{\ell=0}^{2^n-1} \subset \mathbb{R}^{2^n}$ which give the spectral decomposition

$$H = \sum_{\ell=0}^{2^n-1} \lambda_\ell \, |\lambda_\ell\rangle\langle\lambda_\ell|. \tag{S40}$$

Since we have chosen $|\lambda_\ell\rangle$ to be real, $|\lambda_\ell\rangle\langle\lambda_\ell|^T = |\lambda_\ell\rangle\langle\lambda_\ell|$, so $U = e^{-iHt}$ is symmetric (though not typically Hermitian):

$$U^T = \sum_{\ell=0}^{2^n-1} e^{i\lambda_\ell t} \, |\lambda_\ell\rangle\langle\lambda_\ell|^T = U, \tag{S41}$$

therefore satisfying Eq. (4) of the main text.

Time-dependent Hamiltonians can also satisfy this symmetry constraint. Consider more generally a Hamiltonian $H(t)$ and its corresponding propagator $U(t,t')$ from t to t' satisfying $i\partial_{t'} U(t,t') = H(t')U(t,t')$ and $U(t,t) = I$ for all $t \le t'$ in $[0,\tau]$, where τ is the total evolution time. A sufficient condition for $U(0,\tau)$ to satisfy Eq. (4) of the main text is that $H(t)^T = H(t)$ and $H(t) = H(\tau - t)$ for all $t \in [0,\tau]$. (We assume that appropriate continuity in $H(t)$ is assured on physical grounds.)

Proof. For $0 \le t \le \tau/2$ we claim that $U(t, \tau - t) = U(\tau/2, \tau - t)\, U(t, \tau/2)$ is symmetric. Since

$$\big[U(\tau/2, \tau - t)\, U(t, \tau/2)\big]^T = U(t, \tau/2)^T \, U(\tau/2, \tau - t)^T, \tag{S42}$$

it suffices to show that $U(\tau/2, \tau - t)^T$ equals $U(t, \tau/2)$. We will do so by showing that both satisfy the same first-order differential equation with the same initial condition. First:

$$\begin{aligned} i\frac{\partial}{\partial t}U(\tau/2,\tau-t)^T &= -\big[H(\tau-t)\,U(\tau/2,\tau-t)\big]^T \\ &= -U(\tau/2,\tau-t)^T\,H(t). \end{aligned} \tag{S43}$$

Next, differentiating both sides of $U(0,\tau/2) = U(t,\tau/2)\,U(0,t)$ with respect to t gives

$$0 = \left[\frac{\partial}{\partial t}U(t,\tau/2)\right]U(0,t) + U(t,\tau/2)\,\frac{\partial}{\partial t}U(0,t) \tag{S44}$$

and therefore

$$\begin{aligned} \frac{\partial}{\partial t}U(t,\tau/2) &= -U(t,\tau/2)\left[\frac{\partial}{\partial t}U(0,t)\right]U(0,t)^\dagger \\ &= iU(t,\tau/2)\,H(t). \end{aligned} \tag{S45}$$

Since $U(\tau/2,\tau-t)^T$ and $U(t,\tau/2)$, viewed as functions of t, satisfy the same differential equation in Eqs. (S43) and (S45), and both equal I when $t=\tau/2$, they are equal for all $t \in [0,\tau/2]$. Therefore

$$U(t,\tau-t)^T = U(t,\tau/2)^T\,U(\tau/2,\tau-t)^T = U(\tau/2,\tau-t)\,U(t,\tau/2) = U(t,\tau-t). \tag{S46}$$

Setting $t=0$ gives $U(0,\tau)^T = U(0,\tau)$. □

A particularly interesting way to propose MCMC moves that are close in energy but potentially far in Hamming distance is through reverse quantum annealing [53]. That is, by evolving under a Hamiltonian of the form

$$H(t) = \big[1-f(t)\big]\alpha H_{\text{prob}} + f(t)H_{\text{mix}} \tag{S47}$$

where $f:[0,\tau]\to[0,1]$ is an even function about $t=\tau/2$ that starts at $f(0)=0$, gradually ramps up to some maximum value $f(\tau/2)>0$ and then ramps back down symmetrically to $f(\tau)=0$. H_{mix} could be proportional to $\sum_{j=1}^n X_j$ or some other easily-realizable, real matrix. In the limit of $\tau\to\infty$, the final state would be the same as the initial state due to the adiabatic theorem. For finite τ, however, Landau–Zener transitions can occur at avoided crossings. These transitions are typically harmful in adiabatic quantum computing, but in this context they are beneficial: when a small number of them occur, the measured configuration k should be close in energy to the initial one j, but generically far in Hamming distance.

One way to see this is by moving to a rotating frame defined by the eigenvectors of $H(t)$. Let $\{|\lambda_j(t)\rangle\}_{j=0}^{2^n-1}$ be the eigenvectors of $H(t)$, as in Section V A, with corresponding eigenvalues $\{\lambda_j(t)\}_{j=0}^{2^n-1}$, and let

$$V(t) = \sum_{j=0}^{2^n-1} |\lambda_j(t)\rangle\langle j| \tag{S48}$$

be the unitary that changes between the "lab frame" (Schrödinger picture) and a rotating frame (interaction picture). That is, a state $|\psi\rangle$ in the former frame becomes $V^\dagger|\psi\rangle$ in the latter, and an observable O becomes $V^\dagger O V$. (We take this as the mathematical definition of the rotating frame.) Since $V(0)=V(\tau)=I$, we are free to compute transition probabilities in the rotating frame, without ever having to explicitly convert back to the lab frame. The rotating frame dynamics are generated by the Hamiltonian $\tilde{H}(t)$, given by:

$$i\frac{d}{dt}V^\dagger|\psi\rangle = \underbrace{\left(V^\dagger H(t)V + i\dot{V}^\dagger V\right)}_{\tilde{H}(t)}V^\dagger|\psi\rangle. \tag{S49}$$

While $V^\dagger H(t)V = \sum_{j=0}^{2^n-1}\lambda_j(t)|j\rangle\langle j|$ is diagonal, $\tilde{H}(t)$ is generally not, due to the $i\dot{V}^\dagger V$ term. This latter term arises because the eigenbasis of $H(t)$ changes with time, unlike in Section V A. It scales inversely with τ, and is typically highly non-local [54, 55]. In this rotating frame, we clearly see that an initial computational state $|j\rangle$ will always produce a final state $|j\rangle$ (up to a global phase) in the $\tau\to\infty$ limit, as per the adiabatic theorem. When τ is finite, however, $i\dot{V}^\dagger V$ will generate Landau-Zener transitions between instantaneous energy eigenstates. Notice that $\tilde{H}(t)$

here is analogous to $H(\gamma)$ in Section V A, with the diagonal part $V^\dagger H(t) V$ playing the role of H_{prob} and $i\dot{V}^\dagger V$ that of H_{mix}. Invoking the arguments from Section V A, we expect this reverse annealing scheme to generate transitions between configurations that are close in energy but potentially far in Hamming distance.

[1] M. K. Cowles and B. P. Carlin, Markov chain Monte Carlo convergence diagnostics: a comparative review, Journal of the American Statistical Association **91**, 883 (1996).

[2] D. Levin and Y. Peres, *Markov Chains and Mixing Times*, MBK (American Mathematical Society, 2017).

[3] D. Sherrington and S. Kirkpatrick, Solvable model of a spin-glass, Phys. Rev. Lett **35**, 1792 (1975).

[4] A. Callison, N. Chancellor, F. Mintert, and V. Kendon, Finding spin glass ground states using quantum walks, New Journal of Physics **21**, 123022 (2019).

[5] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika **57**, 97 (1970).

[6] R. Chen, J. Liu, and X. Wang, Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications, IEEE Transactions on Signal Processing **50**, 255 (2002).

[7] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, An introduction to MCMC for machine learning, Mach. Learn **50**, 5 (2003).

[8] R. M. Neal, *Probabilistic inference using Markov chain Monte Carlo methods* (Department of Computer Science, University of Toronto Toronto, ON, Canada, 1993).

[9] R. H. Swendsen and J.-S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, Phys. Rev. Lett **58**, 86 (1987).

[10] U. Wolff, Collective Monte Carlo updating for spin systems, Phys. Rev. Lett **62**, 361 (1989).

[11] S. Park, Y. Jang, A. Galanis, J. Shin, D. Stefankovic, and E. Vigoda, Rapid Mixing Swendsen-Wang Sampler for Stochastic Partitioned Attractive Models, in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 54, edited by A. Singh and J. Zhu (PMLR, 2017) pp. 440–449.

[12] A. Barzegar, C. Pattison, W. Wang, and H. G. Katzgraber, Optimization of population annealing Monte Carlo for large-scale spin-glass simulations, Phys. Rev. E **98**, 053308 (2018).

[13] J.-S. Wang, Clusters in the three-dimensional Ising model with a magnetic field, Physica A: Statistical Mechanics and its Applications **161**, 249 (1989).

[14] deLyra, Jorge L., The Wolff algorithm with external sources and boundaries, `http://latt.if.usp.br/cgi-bin-delyra/cntsnd?technical-pages/twawesab/Text.pdf` (2006).

[15] J. Kent-Dobias and J. P. Sethna, Cluster representations and the Wolff algorithm in arbitrary external fields, Phys. Rev. E **98**, 063306 (2018).

[16] J. Houdayer, A cluster Monte Carlo algorithm for 2-dimensional spin glasses, Eur. Phys. J. B **22**, 479 (2001).

[17] B. D. He, C. M. De Sa, I. Mitliagkas, and C. Ré, Scan order in Gibbs sampling: Models in which it matters and bounds on how much, in *Advances in Neural Information Processing Systems*, Vol. 29, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016).

[18] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, Efficient cluster algorithm for spin glasses in any space dimension, Phys. Rev. Lett **115**, 077201 (2015).

[19] P. Virtanen *et al.*, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods **17**, 261 (2020).

[20] D. J. Hsu, A. Kontorovich, and C. Szepesvari, Mixing time estimation in reversible Markov chains from a single sample path, in *Advances in Neural Information Processing Systems*, Vol. 28, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Curran Associates, Inc., 2015).

[21] D. Layden, First-order Trotter error from a second-order perspective, arXiv:2107.08032 (2021).

[22] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient Z gates for quantum computing, Phys. Rev. A **96**, 022330 (2017).

[23] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge Series on Information and the Natural Sciences (Cambridge University Press, 2000).

[24] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, Procedure for systematically tuning up cross-talk in the cross-resonance gate, Phys. Rev. A **93**, 060302 (2016).

[25] N. Sundaresan, I. Lauer, E. Pritchett, E. Magesan, P. Jurcevic, and J. M. Gambetta, Reducing unitary and spectator errors in cross resonance with optimized rotary echoes, PRX Quantum **1**, 020318 (2020).

[26] J. P. T. Stenger, N. T. Bronn, D. J. Egger, and D. Pekker, Simulating the dynamics of braiding of Majorana zero modes using an IBM quantum computer, Phys. Rev. Research **3**, 033171 (2021).

[27] N. Earnest, C. Tornow, and D. J. Egger, Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware, Phys. Rev. Research **3**, 043088 (2021).

[28] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of Trotter error with commutator scaling, Phys. Rev. X **11**, 011020 (2021).

[29] G. C. Knee and W. J. Munro, Optimal Trotterization in universal quantum simulators under faulty control, Phys. Rev. A **91**, 052327 (2015).

[30] S. Endo, Q. Zhao, Y. Li, S. Benjamin, and X. Yuan, Mitigating algorithmic errors in a Hamiltonian simulation, Phys. Rev. A **99**, 012334 (2019).

[31] L. Clinton, J. Bausch, and T. Cubitt, Hamiltonian simulation algorithms for near-term quantum hardware, Nature communications **12**, 1 (2021).

[32] E. van den Berg, Z. K. Minev, and K. Temme, Model-free readout-error mitigation for quantum expectation values, arXiv:2012.09738 (2020).

[33] E. Knill, Fault-tolerant postselected quantum computation: Threshold analysis, quant-ph/0404104 (2004).

[34] J. J. Wallman and J. Emerson, Noise tailoring for scalable quantum computation via randomized compiling, Phys. Rev. A **94**, 052325 (2016).

[35] Y. Kim, C. J. Wood, T. J. Yoder, S. T. Merkel, J. M. Gambetta, K. Temme, and A. Kandala, Scalable error mitigation for noisy quantum circuits produces competitive expectation values, arXiv:2108.09197 (2021).

[36] A. W. Cross, A. Javadi-Abhari, T. Alexander, N. de Beaudrap, L. S. Bishop, S. Heidel, C. A. Ryan, J. Smolin, J. M. Gambetta, and B. R. Johnson, OpenQASM 3: A broader and deeper quantum assembly language, arXiv:2104.14722 (2021).

[37] A. D. Córcoles, M. Takita, K. Inoue, S. Lekuch, Z. K. Minev, J. M. Chow, and J. M. Gambetta, Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits, Phys. Rev. Lett. **127**, 100501 (2021).

[38] A. H. Bowker, A test for symmetry in contingency tables, Journal of the American Statistical Association **43**, 572 (1948).

[39] Q. McNemar, Note on the sampling error of the difference between correlated proportions or percentages, Psychometrika **12**, 153 (1947).

[40] A. Krampe and S. Kuhnt, Bowker's test for symmetry and modifications within the algebraic framework, Computational statistics & data analysis **51**, 4124 (2007).

[41] G. E. Cho and C. D. Meyer, Comparison of perturbation bounds for the stationary distribution of a Markov chain, Linear Algebra and its Applications **335**, 137 (2001).

[42] StataCorp, Stata manual: symmetry, `https://www.stata.com/manuals/rsymmetry.pdf`.

[43] A. Gilchrist, N. K. Langford, and M. A. Nielsen, Distance measures to compare real and ideal quantum processes, Phys. Rev. A **71**, 062310 (2005).

[44] J. J. Sakurai and J. Napolitano, *Modern quantum mechanics; 2nd ed.* (Addison-Wesley, San Francisco, CA, 2011).

[45] E. Farhi and S. Gutmann, Analog analogue of a digital quantum computation, Phys. Rev. A **57**, 2403 (1998).

[46] S. Blanes, F. Casas, J. Oteo, and J. Ros, The Magnus expansion and some of its applications, Physics Reports **470**, 151 (2009).

[47] D. Zueco, G. M. Reuther, S. Kohler, and P. Hänggi, Qubit-oscillator dynamics in the dispersive regime: Analytical theory beyond the rotating-wave approximation, Phys. Rev. A **80**, 033846 (2009).

[48] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, Exponential algorithmic speedup by a quantum walk, in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03 (Association for Computing Machinery, New York, NY, USA, 2003) p. 59–68.

[49] K. Kechedzhi, V. Smelyanskiy, J. R. McClean, V. S. Denchev, M. Mohseni, S. Isakov, S. Boixo, B. Altshuler, and H. Neven, Efficient Population Transfer via Non-Ergodic Extended States in Quantum Spin Glass, in *13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 111, edited by S. Jeffery (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018) pp. 9:1–9:16.

[50] V. N. Smelyanskiy, K. Kechedzhi, S. Boixo, S. V. Isakov, H. Neven, and B. Altshuler, Nonergodic delocalized states for efficient population transfer within a narrow band of the energy landscape, Phys. Rev. X **10**, 011017 (2020).

[51] V. N. Smelyanskiy, K. Kechedzhi, S. Boixo, H. Neven, and B. Altshuler, Intermittency of dynamical phases in a quantum spin glass, arXiv:1907.01609 (2019).

[52] C. L. Baldwin and C. R. Laumann, Quantum algorithm for energy matching in hard optimization problems, Phys. Rev. B **97**, 224201 (2018).

[53] E. Crosson and D. Lidar, Prospects for quantum enhancement with diabatic quantum annealing, Nature Reviews Physics , 1 (2021).

[54] D. Sels and A. Polkovnikov, Minimizing irreversible losses in quantum systems by local counterdiabatic driving, Proceedings of the National Academy of Sciences **114**, E3909 (2017).

[55] M. Kolodrubetz, D. Sels, P. Mehta, and A. Polkovnikov, Geometry and non-adiabatic response in quantum and classical systems, Physics Reports **697**, 1 (2017).