Supplementary Information for Polynomially efficient quantum enabled variational Monte Carlo for training neural-network quantum states for physico-chemical applications

Manas Sajjan[†],[1] Vinit Singh[†],[1] and Sabre Kais[1,*]

[1]*Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606*

CONTENTS

* skais@ncsu.edu

† These authors contributed equally to this work

S1. DEFINITIONS/NOTATIONS

We re-iterate the definitions considered in the main text and to be used subsequently for remaining parts of the Supplementary Information.

- $\vec{v} = (v_1, v_2, \ldots, v_n)$: A binary configuration of the n visible units, where each v_i is a binary variable taking values in $\{-1, 1\}$. Each configuration represents a possible classical state of the graph G_2 or the visible set of neurons of graph G_1. Unless otherwise specified, we shall sometimes omit the vector sign above and simply use $v \in [0, \cdots, 2^n - 1]$. This is akin to representing the corresponding integer index for the configuration in binary (with -1 substituted by 0).

- $\mathbb{V} = \{\vec{v} | \vec{v} \in \{-1, 1\}^n\}$. One must note $|\mathbb{V}| = 2^n$.

- $\rho^v_{v'}$: The matrix element of the state $\rho(\vec{X})$ corresponding to basis states v and v'. It represents the density matrix of the state.

$$\rho^v_{v'}(\vec{X}) \propto exp\left(-\beta \sum_i a_i v_i + \sum_i a_i^* v'_i\right) \prod_{j=1}^{m} \Gamma_j(\vec{v}, \vec{b}, \vec{W})\, \Gamma_j(\vec{v}', \vec{b^*}, \vec{W}^*) \tag{S1}$$

  where, $\Gamma_j(\vec{v}, \vec{b}, \vec{W}) = \cosh\left[\beta\left(b_j + \sum_{i=1}^n W_{ij} v_i\right)\right]$ and

  - a_i: A complex-valued parameter associated with the bias for the i-th visible unit in the system.
  - b_j: A complex-valued parameter associated with the bias for the j-th hidden unit in the system.
  - W_{ij}: The weight (coupling) between the i-th visible unit and j-th hidden unit.

- $\vec{X}$: A shorthand representation for all the parameters $(\vec{a}, \vec{b}, \vec{W})$ that define the state $\rho(\vec{X})$. These include complex-valued parameters associated with interactions and biases in the network.

- β: The inverse temperature parameter.

- $H^{v'}_v$: The Hamiltonian matrix element of the driver (system being studied) between state configurations v and v'.

- The diagonal elements, ρ^v_v, are real and satisfy the normalization condition $\sum_v \rho^v_v = 1$. They can be seen as a probability distribution.

$$\rho^v_v(\vec{x}) \;\propto\; exp\left(-2\beta \sum_i \mathrm{Re}(a_i) v_i\right) \prod_{j=1}^{m} \left|\cosh\left(b_j + \sum_{i=1}^{n} W_{ij} v_i\right)\right|^2 \tag{S2}$$

- $E_{\mathrm{loc}}(v) = \frac{\sum_{v'} \rho^v_{v'} H^{v'}_v}{\rho^v_v}$: The local energy of the driver for the configuration v. It measures the effective energy associated with the state v, computed by taking into account all the other configurations v' to which it is connected via the Hamiltonian matrix element of the driver.

- $\phi(v)$: The distribution that represents the state of the surrogate network. It is defined as

$$\phi_{\vec{v}}(\vec{l}(\vec{X}), \vec{J}(\vec{X})) \;\propto\; e^{-\beta \sum_i l_i(\vec{X}) v_i + \sum_{ij} J_{ij}(\vec{X}) v_i v_j} \tag{S3}$$

  Here $l_i \in (\vec{l}(\vec{X})$ denotes the on-site field at each spin of the surrogate network G_2 in main text and $J_{ij}(\vec{X}) \in \vec{J}(\vec{X}))$ denotes the mutual coupling between a pair of spins of G_2. $\phi(v)$ satisfies the normalization condition $\sum_v \phi(v) = 1$.

- $\kappa(v)$: A configuration dependent pre-factor chosen to establish the equivalence between ρ^v_v and $\phi(v)$, as defined above

$$\rho^v_v \approx \kappa(v)\phi(v)$$

S2. ALTERATION IN VARIANCE OF THE SAMPLED ESTIMATE OF THE LOCAL ENERGY DUE TO PARAMETERIZATION OF $\kappa(\vec{v})$

We would like to prove in this section that certain choices/parameterization of $\kappa(\vec{v})$ introduced in previous section can lead to reduction in variance of the sampled estimate of a random variable. However, before we delve into that we first show that the mean of the random variable using samples generated from the surrogate distribution of $\phi(\vec{v})$ is an unbiased estimator of the target sample estimate computed using the original distribution $P(\vec{v}) = \rho_v^v(\vec{X})$ (see main text). In other words, mean of a random variable of interest($E_{loc}(v)$) over the distribution ρ_v^v is equivalent to the mean of $E_{loc}(v)\kappa(v)$ over the distribution $\phi(v)$. We do this by using the following lemma

Lemma 1. ***Means obtained from*** ρ_v^v ***and*** $\phi(v)$*: The mean obtained by averaging* $E_{loc}(v)$ *over probability distribution* ρ_v^v *is the same as the mean of* $E_{loc}(v)\kappa(v)$ *over probability distribution* $\phi(v)$*.*

$$\langle H \rangle = \langle E_{loc}(v) \rangle_{\rho_v^v} = \langle E_{loc}(v)\kappa(v) \rangle_{\phi(v)}$$

Proof.

$$\begin{aligned}
\langle H \rangle &= Tr(\rho H) \\
&= \sum_v \rho_v^v \Big(\frac{\sum_{v'} \rho_{v'}^v H_v^{v'}}{\rho_v^v} \Big) \\
&= \sum_v \rho_v^v E_{loc}(v) = \langle E_{loc}(v) \rangle_{\rho_v^v}
\end{aligned} \tag{S4}$$

$$\begin{aligned}
&\langle E_{loc}(v)\kappa(v) \rangle_{\phi(v)} \\
&= \sum_v E_{loc}(v)\kappa(v)\phi(v) \\
&= \sum_v E_{loc}(v)\rho_v^v \\
&= \langle E_{loc}(v) \rangle_{\rho_v^v}
\end{aligned} \tag{S5}$$

Thus from S4 and S5,

$$\boxed{\langle H \rangle = \langle E_{loc}(v) \rangle_{\rho_v^v} = \langle E_{loc}(v)\kappa(v) \rangle_{\phi(v)}} \tag{S6}$$

□

Thus we see we have access to two distributions ρ_v^v and $\phi(v)$. If we sample v from either and compute the mean of two different random variables, i.e., $E_{loc}(v)$ for ρ_v^v and $E_{loc}(v)\kappa(v)$ for $\phi(v)$, then the population mean of the two would be equal. The mean of either of these two random variables can act as a proxy of the other. One must note that practically since ρ_v^v is obtained from fitting, then $\rho_v^v = \kappa(v)\phi(v) + \epsilon(v)$, where $\epsilon(v)$ is the fitting error for each configuration, then

$$\langle E_{\rm loc}(v)\kappa(v) \rangle_\phi = \langle E_{\rm loc}(v) \rangle_\rho - \sum_v E_{\rm loc}(v)\epsilon(v).$$

Hence the estimator is unbiased iff $\sum_v E_{\rm loc}(v)\epsilon(v) = 0$ (in particular if $\epsilon \equiv 0$) i.e. in the limit of low fitting errors which is the regime we operate in.

Now we would like to prove the main theorem in this Section

Theorem S2.1. ***Variances obtained from*** ρ_v^v ***and*** $\phi(v)$***:*** *For a specific form of* $\kappa(v) = \frac{1}{\lambda |E_{loc}(v)|^\alpha}$*, the difference between the variance of the sampled random variable* $E_{loc}(v)$ *over* ρ_v^v *and the variance of* $E_{loc}(v)\kappa(v)$ *over* $\phi(v)$ *is given by,*

$$\sigma^2_{E_{loc}(v)} - \sigma^2_{E_{loc}(v)\kappa(v)} = Cov\left(|E_{loc}(v)|^{2-\alpha}, |E_{loc}(v)|^\alpha \right)$$

Proof.

$$\begin{aligned}
\sigma^2_{E_{loc}(v)} &= \sum_v E^2_{loc}(v)\rho_v^v - \langle E_{loc}(v) \rangle^2_{\rho_v^v} \\
\sigma^2_{E_{loc}(v)\kappa(v)} &= \sum_v E_{loc}(v)\kappa(v)^2\phi(v) - \langle E_{loc}(v)\kappa(v) \rangle^2_{\phi(v)}
\end{aligned}$$

We denote the sample mean in each case as: $\bar{E}_\rho = \frac{1}{N_s}\sum_{t=1,v(t)\sim\rho}^{N_s} E_{\rm loc}(v^{(t)})$ and $\bar{E}_\phi = \frac{1}{N_s}\sum_{t=1,v^{(t)}\sim\phi}^{N_s} E_{\rm loc}(v^{(t)})\kappa(v^{(t)})$. We need to see how the sample means differs from the population mean for a fixed number of samples in each N_s in each case. Note even though the respective means are the same by construction (see Eq. S6), but the random variable in each case is not the same. The difference between the sample and true/population mean for each of the random variable can be quantified using Chebyshev inequality where the variance makes an appearance as follows:

$$P(|\bar{E}_\phi - \langle E_{loc}(v)\kappa(v)\rangle_{\phi(v)}| \geq \epsilon) \leq \frac{\mathrm{Var}(E_{loc}(v)\kappa(v))}{N_s\epsilon^2} \tag{S7}$$

$$P(|\bar{E}_\rho - \langle E_{loc}(v)\rangle_{\rho_v^v}| \geq \epsilon) \leq \frac{\mathrm{Var}(E_{loc}(v))}{N_s\epsilon^2} \tag{S8}$$

where $\mathrm{Var}(E_{loc}(v)\kappa(v)) = \sigma^2_{E_{loc}(v)\kappa(v)}$ and $\mathrm{Var}(E_{loc}(v)) = \sigma^2_{E_{loc}(v)}$. We need to see for the same N_s if the variances $\mathrm{Var}(E_{loc}(v)\kappa(v))$, $\mathrm{Var}(E_{loc}(v))$ are equivalent or not?

$$\begin{aligned}
&\text{Thus, } \mathrm{Var}(E_{loc}(v))_{\rho_v^v} - \mathrm{Var}(E_{loc}(v)\kappa(v))_{\phi(v)} \\
&= \sigma^2_{E_{loc}(v)} - \sigma^2_{E_{loc}(v)\kappa(v)} \\
&= \sum_v E^2_{loc}(v)\rho_v^v - \sum_v (E_{loc}(v)\kappa(v))^2\phi(v) + \langle E_{loc}(v)\kappa(v)\rangle^2_{\phi(v)} - \langle E_{loc}(v)\rangle^2_{\rho_v^v} \\
&= \sum_v E^2_{loc}(v)\rho_v^v - \sum_v (E_{loc}(v)\kappa(v))^2\phi(v) \\
&= \sum_v E^2_{loc}(v)\rho_v^v - \sum_v E^2_{loc}(v)\kappa^2(v)\phi(v) \\
&= \sum_v E^2_{loc}(v)\rho_v^v - \sum_v \frac{E^2_{loc}(v)\left(\rho_v^v\right)^2}{\phi(v)} \\
&= \sum_v E^2_{loc}(v)\rho_v^v \left(1 - \frac{\rho_v^v}{\phi(v)}\right)
\end{aligned}$$

Now if we assume $\boxed{\kappa(v) = \dfrac{1}{\lambda|E_{loc}(v)|^\alpha} \;\Rightarrow\; \phi(v) = \lambda|E_{loc}(v)|^\alpha\rho_v^v}$

$$\begin{aligned}
\text{then} \quad \lambda &= \frac{\sum_v \phi(v)}{\sum_v |E_{loc}(v)|^\alpha \rho_v^v} = \frac{1}{\sum_v |E_{loc}(v)|^\alpha \rho_v^v} \\
\text{or} \quad \phi(v) &= \frac{|E_{loc}(v)|^\alpha \rho_v^v}{\sum_{v'} |E_{loc}(v')|^\alpha \rho_{v'}^{v'}} \\
\text{or} \quad \frac{\rho_v^v}{\phi(v)} &= \frac{\sum_{v'} |E_{loc}(v')|^\alpha \rho_{v'}^{v'}}{|E_{loc}(v)|^\alpha}
\end{aligned}$$

In other words, we have

$$
\begin{aligned}
&\sigma^2_{E_{loc}(v)} - \sigma^2_{E_{loc}(v)\kappa(v)} \\
&= \sum_v E^2_{loc}(v)\rho^v_v \left(1 - \frac{\sum_{v'} |E_{loc}(v')|^\alpha \rho^{v'}_{v'}}{|E_{loc}(v)|^\alpha}\right) \\
&= \sum_v E^2_{loc}(v)\rho^v_v \left(1 - \frac{\langle |E_{loc}(v)|^\alpha \rangle_{\rho^v_v}}{|E_{loc}(v)|^\alpha}\right) \\
&= \sum_v E^2_{loc}(v)\rho^v_v - \left(\sum_v \frac{E^2_{loc}(v)\rho^v_v}{|E_{loc}(v)|^\alpha}\right) \langle |E_{loc}(v)|^\alpha \rangle_{\rho^v_v} \\
&= \left\langle |E_{loc}(v)|^2 \right\rangle_{\rho^v_v} - \left\langle |E_{loc}(v)|^{2-\alpha} \right\rangle_{\rho^v_v} \langle |E_{loc}(v)|^\alpha \rangle_{\rho^v_v} \\
&= \mathrm{Cov}\left(|E_{loc}(v)|^{2-\alpha}, |E_{loc}(v)|^\alpha\right)
\end{aligned}
$$

Or

$$
\boxed{\sigma^2_{E_{loc}(v)} - \sigma^2_{E_{loc}(v)\kappa(v)} = \mathrm{Cov}\left(|E_{loc}(v)|^{2-\alpha}, |E_{loc}(v)|^\alpha\right)} \tag{S9}
$$

□

Lemma 2. *Let's define*

$$
Z = |E_{loc}(v)|^\alpha \qquad \textit{and} \qquad g(Z) = |E_{loc}(v)|^{2-\alpha} = Z^{2/\alpha - 1}
$$

Such that from Eq.S9,

$$
\sigma^2_{E_{loc}(v)} - \sigma^2_{E_{loc}(v)\kappa(v)} = Cov\left(Z^{2/\alpha-1}, Z\right) = Cov\left(g(Z), Z\right)
$$

Given $Z \geq 0$, then show that $Cov(g(Z), Z) > 0$ if $g(Z)$ is non-decreasing, i.e., $g'(Z) \geq 0$, provided $E[|Z|] < \infty$ and $E[|g(Z)|] < \infty$.

Proof.

$$
\begin{aligned}
\mathrm{Cov}(g(Z), Z) &= \mathbb{E}[(g(Z) - \langle g(Z)\rangle)(Z - \langle Z\rangle)] \\
&\quad \text{This expectation and } \langle\rangle \text{ is over } \rho^v_v \text{ as is the covariance.} \\
&= \mathbb{E}\left[(Z - \langle Z\rangle)(g(Z) + g(\langle Z\rangle) - g(Z) - \langle g(Z)\rangle)\right] \\
&= \mathbb{E}[(Z - \langle Z\rangle)(g(Z) - g(\langle Z\rangle))] + \mathbb{E}[(Z - \langle Z\rangle)(g(\langle Z\rangle) - \langle g(Z)\rangle)] \\
&= \mathbb{E}[(Z - \langle Z\rangle)(g(Z) - g(\langle Z\rangle))] + \mathbb{E}[(Z - \langle Z\rangle)(g(\langle Z\rangle) - \langle g(Z)\rangle)]
\end{aligned}
$$

$g(\langle Z\rangle) - \langle g(Z)\rangle$ is independent of Z and hence can be taken outside expectation. This is because we have g at $\langle Z\rangle$ and $\langle g(Z)\rangle$, both independent of Z now and just scalars.

$$
\begin{aligned}
&\Rightarrow \mathbb{E}[(Z - \langle Z\rangle)(g(Z) - g(\langle Z\rangle))] + (g(\langle Z\rangle) - \langle g(Z)\rangle)\mathbb{E}[(Z - \langle Z\rangle)] \\
&\qquad\qquad \text{As} \quad \mathbb{E}[(Z - \langle Z\rangle)] = 0 \\
&\Rightarrow \mathbb{E}[(Z - \langle Z\rangle)(g(Z) - g(\langle Z\rangle))]
\end{aligned}
$$

Now as $g(Z)$ is a non-decreasing function,

$$
\begin{aligned}
&\frac{g(Z_1) - g(Z_2)}{Z_1 - Z_2} \geq 0 \qquad \text{(follows from derivative itself)} \\
\Rightarrow &(Z_1 - Z_2)^2 \frac{g(Z_1) - g(Z_2)}{Z_1 - Z_2} \geq 0 \\
\Rightarrow &(g(Z_1) - g(Z_2))(Z_1 - Z_2) \geq 0
\end{aligned}
$$

Thus,

$$\mathbb{E}[(g(Z) - g(\langle Z \rangle))(Z - \langle Z \rangle)] \geq 0$$

where $Z_1 \to Z$, $Z_2 \to \langle Z \rangle$.

Now if we take $Z_1 = Z$ and $Z_2 = \langle Z \rangle$, then that means

$$\mathbb{E}[(g(Z) - g(\langle Z \rangle))(Z - \langle Z \rangle)] \geq 0$$

So,

$$\boxed{\text{Cov}(g(Z), Z) \geq 0}$$

□

Lemma 3. *In the range* $0 < \alpha \leq 2$*, show that*

$$\sigma^2_{E_{loc}(v)} - \sigma^2_{E_{loc}(v)\kappa(v)} = Cov(Z^{2/\alpha - 1}, Z) \geq 0$$

Proof. When $0 < \alpha \leq 2$ then $g(Z)$ is non-decreasing as $g'(Z) = \frac{(2/\alpha - 1)}{Z} Z^{2/\alpha - 1} \geq 0 \quad \because Z \geq 0$ Thus by the assertions of the above lemma $\sigma^2_{E_{loc}(v)} - \sigma^2_{E_{loc}(v)\kappa(v)} = \text{Cov}(Z^{2/\alpha - 1}, Z) = \text{Cov}(g(Z), Z) \geq 0$. Similarly when $\alpha > 2$, $Z^{2/\alpha - 1}$ is a non-increasing function as $g'(Z) = \frac{(2/\alpha - 1)}{Z} Z^{2/\alpha - 1} \leq 0$ as $Z \geq 0$. If that is the case, by the exact same logic in the last lemma, we can say $\text{Cov}(g(Z), Z) \leq 0$
Also, $\alpha \leq 0$ $g(Z) = Z^{2/\alpha - 1}$ is still non-increasing as $g'(Z) = \frac{(2/\alpha - 1)}{Z} Z^{2/\alpha - 1} \leq 0 \quad \because Z \geq 0$. Thus even then $\text{Cov}(g(Z), Z) \leq 0$
So we have a full characterization:

$$\sigma^2_{E_{loc}(v)} - \sigma^2_{E_{loc}(v)\kappa(v)} = \text{Cov}(|E_{loc}(v)|^{2-\alpha}, |E_{loc}(v)|^{\alpha}) = \text{Cov}(Z^{2/\alpha - 1}, Z), \quad (Z = |E_{loc}(v)|^{\alpha} \geq 0)$$

$\alpha < 0$	$0 \leq \alpha \leq 2$	$\alpha > 2$
$Z^{2/\alpha - 1}$ is non-increasing $\text{Cov}(Z^{2/\alpha - 1}, Z) \leq 0$	$Z^{2/\alpha - 1}$ is non-decreasing $\text{Cov}(Z^{2/\alpha - 1}, Z) \geq 0$	$Z^{2/\alpha - 1}$ is non-increasing $\text{Cov}(Z^{2/\alpha - 1}, Z) \leq 0$

Special points:

$\alpha = 1$	$\alpha = 2$
$\text{Cov}(Z, Z) \geq 0$	$\text{Cov}(1, Z) = 0$

□

S3. DATA DRIVEN CONSTRUCTION OF THE SURROGATE PROBABILITY DISTRIBUTION

In the previous sections, we have shown that any probability distribution can be written in terms of an exponential of an arbitrary-degree polynomial function of the spin configuration (see Theorem 1.1 in main manuscript or Theorem 3.1 in Section S2). So far we have proven that it is always possible to represent an arbitrary probability distribution $P(\vec{v}) : \mathbb{V} \rightarrow [0,1]$ as $\mathcal{N}\kappa(\vec{v})\phi(\vec{v})$ where $\phi(\vec{v})$ is the distribution of the surrogate network defined as

$$\phi(\vec{v},\vec{X}) \propto exp\left(-\beta \sum_i l_i(\vec{X})v_i + \sum_{ij} J_{ij}(\vec{X})v_i v_j + \cdots\right)$$

Here $l_i \in \vec{l}(\vec{X})$ denotes the on-site field at each spin of the surrogate network and $J_{ij}(\vec{X}) \in \vec{J}(\vec{X})$ denotes the mutual coupling between a pair of spins of the surrogate. The kernel function $\kappa(\vec{v})$ accounts for additional flexibility in the construction when the surrogate distribution is truncated to a finite degree k. In this section we provide a concrete data-driven recipe to construct the same. Here we choose $P(\vec{v}) = \rho_v^v$ where ρ_v^v is the probability distribution associated with graph G_1 in main text. The surrogate so defined (see graph G_2 in main text) has been truncated to second-degree i.e. $k = 2$. Even though the algorithm we describe below works for any value of k, however, it should be noted that implementing a polynomial of degree k requires k-qubit gates in a quantum circuit, and hence a large k can increase the complexity of the circuit. Therefore, one would prefer to work with low-degree polynomials for practical implementation which motivates the choice for $k = 2$. At the end of this protocol we shall have a decomposition of the following kind

$$\rho_v^v(v,\vec{X},\beta) = \phi(v,\vec{X})\kappa(v,\vec{X},\beta) \ + \epsilon(v)$$

where $\epsilon(\vec{v})$ is the fitting error associated with the approximation $\rho_v^v \approx \phi(v)\kappa(v)$.

A. Sampling configurations for fitting

To ensure that the surrogate distribution closely approximates the actual distribution, we aim to fit the parameters of the polynomial function through a non-linear fitting process. Ideally, we would use all possible configurations to fit these parameters. However, this approach would be computationally prohibitive and moot the advantage offered by our algorithm. Therefore, we strategically select a sample of configurations to perform the fitting. The selected configurations include:

- **The configurations with largest ρ_v^v value**: These configurations are important because they are the most probable and thus contribute significantly to the overall distribution. We can find these configurations efficiently using the algorithm outlined in Section S3 B. Selecting these ensures that the model accurately represents the high-probability regions of the distribution.

- **Random configurations**, which represent the bulk of the less probable configurations. Incorporating these ensures the model generalizes well and prevents overfitting to the most probable configurations alone.

Focusing on the most probable configurations is crucial because they dominate the sampling process; these configurations have the highest ρ_v^v values and will appear most frequently during sampling. When fitting the surrogate distribution $\phi(v)$, we need to ensure it performs well for these configurations since they represent the most critical part of the probability landscape. However, there is a risk associated with only using the best configurations: overfitting. If the model fits exclusively to high-probability configurations, it may perform poorly on low-probability ones. In extreme cases, this could cause the model to predict low-probability configurations so inaccurately that they interfere with high-probability predictions, compromising the overall accuracy. [See Fig S1] This is why we include random configurations in the sample. These configurations, though less probable, help to ensure that the surrogate distribution generalizes well across the entire probability space. They prevent the model from overfitting to the high-probability regions and ensure that $\phi(v)$ adequately represents both high- and low-probability configurations. By constructing the surrogate network with this balanced set of configurations—both the most probable and a representative sample of lower-probability ones—we create a robust model that maintains accuracy across the full distribution without being biased toward a narrow set of configurations. We keep the split between the best configuration and random configuration as a hyperparameter. In this work, we maintain a ratio of 25% best configurations and 75% random configurations. The total number of samples is kept below $O(n^2)$.

B. Algorithm for finding q-largest configurations

Given a probability distribution ρ_v^v which has its support over $O(2^n)$ spin configurations as $\vec{v} \in \mathbb{V}$. We need to find q-spin configurations that maximize ρ_v^v which from the main text is defined as: $\rho_v^v = \exp\Big(-2\beta \sum_{i=1}^n Re(a_i)v_i\Big) \prod_{j=1}^m |\cosh(b_j + \sum_{i=1}^n w_{ij}v_i)|^2$ We exploit the structure of the probability distribution to find the q-largest configurations. Note that the first exponential term is maximized for a configuration $\{v_i\}^{(\vec{0})} = \{-\text{sgn}(a_i)\}$ for $i = 1, \ldots, n$. Individual cosh terms are maximized when $\{v_i\}^{(j)} = \text{sgn}(\text{Re}(w_{ij}))$ or $\{v_i\}^{(m+j)} = -\text{sgn}(\text{Re}(w_{ij}))$ for $j = 1, \ldots, m$. Starting from an initial set of 2m+1 candidates, we run an iterative algorithm that perturbs these configurations to identify the approximate q-largest configurations.

Algorithm Steps

1. Initialize a list **config** to store the candidate spin configurations and another list **result** to store the spin configuration with the largest ρ_v^v.
2. Choose {-sgn(Re(a)), sgn(Re(w_j)), and -sgn(Re(w_j))} as the first (2m+1) contenders to the largest spin configurations and store them in **config**. These seeds are chosen because they individually maximize different terms in ρ_v^v.
3. Compute ρ_v^v (upto normalization) for each (2m+1) configuration, sort them in descending order of their ρ_v^v, and remove any duplicates.
4. Run q iterations:
   (a) Select a configuration $v_i \in$ **config** with the largest ρ_v^v and store it in **result**. Remove it from **config**.
   (b) Perform single-site perturbations on v_i, flipping each spin to generate n new candidate configurations $\{v_i^j\}_{j=1}^n$.
   (c) For each new configuration v_i^j not already in config, compute $\rho_v^v(v_i^j)$ and merge it into **config** based on its $\rho_v^v(v_i^j)$ values.
   (d) Trim **config** to maintain only the top q candidates by discarding configurations with small $\rho_v^v(v_i^j)$ values.
5. Return **result** - the approximate q-best configurations.

Time Complexity: The algorithm runs in $O(qn)$. At each iteration, we generate $O(n)$ new configurations and insert each into a sorted list. Depending on the choice of data structure, the insertion operation can take $O(\log(q))$ time for an array or can be done in $O(1)$ time by having a heap implementation. The nature of the algorithm is heuristic and works well with high probability for general parameter sets we dealt with so far. However, it might be possible to curate special instances where it might fail to obtain the best k configurations .

C. Final Fitting Algorithm

We achieve the best fit using a two-fold optimization protocol. First, we fit the logarithm of the ρ_v^v distribution to a polynomial model through a weighted least-squares (LSQ) fitting procedure. Then, we refine the results using non-linear optimization techniques, such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. This two-step approach is necessary because the optimization landscape is highly non-convex, making the results of non-linear optimization highly sensitive to the initial parameter values. The LSQ fitting step provides a robust starting point by converging to an optimal solution with high reliability. This initial solution is then used as input for the non-linear optimization step, which fine-tunes the parameters to achieve a more accurate fit. This meticulous process is crucial because the fitting procedure plays an essential role in constructing the surrogate network. Ensuring that this mapping is as precise as possible is fundamental to the success of the entire framework.

Polynomial fitting: We transform the above non-linear fitting problem into a polynomial fitting problem by taking a logarithm of both sides.

$$\log\rho_v^v(v, \vec{X}) \approx c_0 + \sum_i l_i(\vec{X})v_i + \sum_{i,j} J_{ij}(\vec{X})v_i v_j + \cdots$$

The polynomial to be obtained by fitting has a constant term c_0, a linear term involving v_i, with coefficients $l_i(\vec{X})$, and quadratic terms involving $v_i v_j$, with coefficients $J_{ij}(\vec{X})$. Note that we restrict our discussion to polynomials with up to quadratic terms, but the method below is general for any order of polynomials.

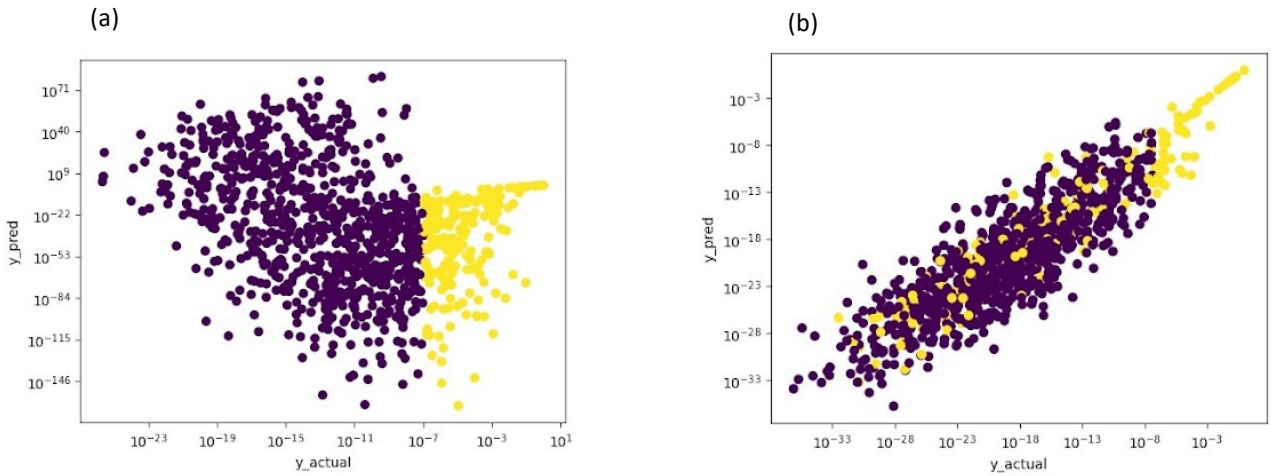

FIG. S1. **The figure demonstrates how the choice of configuration significantly impacts the quality of the model's predictions**. In both plots, yellow points represent configurations used during fitting, while blue points denote configurations not included in the fitting process and are unseen to the model. An ideal fitting would be seen as a straight line with the prediction values (y_{pred}) exactly equal to the actual values (y_{actual}). (a) The left figure shows result of fitting performed using only the top configurations. While the model predicts fairly well for the selected configurations (yellow points), it fails dramatically for unseen configurations (blue points), especially the prediction for lower value points exceed even the high value points in some cases. (b) In the right Figure, we incorporate a mix of random configurations alongside the top configurations, and it can be seen that the model maintains accurate predictions for high values (large y_{actual}) while keeping small values in check. Also note how the algorithm for selecting the best configurations in Section S3 B ensures that all configurations with large y_{actual} values are chosen accurately, as evidenced by the consistent presence of yellow points on the far-right side of the plots.

We aim to find the optimal values for c_0, l_i, and J_{ij} from samples ($\{v^i\}$) drawn from the distribution $\rho_v^v(v)$ as mentioned in Section S3 B. However, we want to avoid having small values of $\rho_v^v(v)$ (which leads to highly negative log values) dominate the fitting process. Therefore, we perform a weighted fit, where the weights are proportional to $\rho_v^v(v)$. This gives more influence to higher values of $\rho_v^v(v)$ during the fitting process. The fitting process solves a weighted least squares (lsq) problem to find the coefficients c_0, l_i, and J_{ij} that best approximate the function. Mathematically, we aim to minimize the weighted sum of squared residuals:

$$\min_{\theta} \|T(A\theta - Y)\|_2$$

Where:

- $\theta = [c_0, l_1, l_2, \ldots, l_N, J_{11}, J_{12}, \ldots, J_{NN}]$ represents the vector of coefficients (parameters).
- Y is the vector of actual values $\{\log \rho_v^v(v^{(i)})\}$ for the sample set $\{v^{(i)}\}$.
- T is a diagonal weight matrix, where $T_{ii} = \sqrt{\rho_v^v(v^{(i)})}$.

The matrix A is the design matrix that contains the constant, linear, and quadratic terms. Its structure is:

$$A = \begin{bmatrix} 1 & v_1^{(1)} & v_2^{(1)} & \ldots & v_N^{(1)} & (v_1^{(1)})^2 & v_1^{(1)}v_2^{(1)} & \ldots & (v_N^{(1)})^2 \\ 1 & v_1^{(2)} & v_2^{(2)} & \ldots & v_N^{(2)} & (v_1^{(2)})^2 & v_1^{(2)}v_2^{(2)} & \ldots & (v_N^{(2)})^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 1 & v_1^{(k)} & v_2^{(k)} & \ldots & v_N^{(k)} & (v_1^{(k)})^2 & v_1^{(k)}v_2^{(k)} & \ldots & (v_N^{(k)})^2 \end{bmatrix}$$

The solution to this problem is given by:

$$\theta = (A^T T A)^{-1} A^T T Y$$

Non-linear fitting: The optimal values for the parameters $\theta = \{c_0, l_i, J_{ij}\}$, obtained from the preceding least-squares optimization, serve as a starting point for the non-linear optimization procedure. This step refines the parameter estimates by directly minimizing the difference between the distributions, yielding more accurate results, in contrast to the previous step, where the logarithms of the distributions were compared.

The cost function for this optimization is defined as:

$$C = \min_{\theta} \left\| \exp(P_k(\theta, \upsilon)) - \rho_{\upsilon}^{\upsilon}(\upsilon) \right\|_2$$

where $\exp(P_k(\theta, \upsilon))$ represents the approximate probability distribution represented by the exponential of the Polynomial dependent on the parameters θ, and $\rho_{\upsilon}^{\upsilon}(\upsilon)$ is the target distribution. The goal is to minimize the L_2-norm of the difference between these distributions. In our work, we employ the BFGS algorithm, a quasi-Newton optimization method, to minimize the cost function. BFGS uses both gradient information and an approximation of the Hessian matrix to guide the search for the optimal parameters. Gradients are approximated using finite differences, allowing BFGS to iteratively refine the parameters despite the absence of explicit derivatives. The result of the non-linear fitting provides the optimal values for the parameters c_0, l_i, and J_{ij}. These parameters can then be used to construct the values of $\phi_{\upsilon}(\vec{X})$ for any given configuration υ.

Full runtime complexity of this step for the benchmarks used: If we review this surrogate-fitting procedure, it is clear that it consists of three steps: (i) selection of fitting configurations via a top-q search combined with additional random samples, (ii) a weighted least-squares (LSQ) polynomial regression to initialize the surrogate parameters, and (iii) non-linear refinement using BFGS. The top-q search scales as $O(qn)$ per iteration, where q is the number of retained configurations and is kept polynomial in system size, specifically $q = O(n^2) \ll 2^n$. In all benchmarks we use $q \approx 2n^2$, with a fixed split of 25% top-probability and 75% random samples, corresponding to $q \approx 128$ for $n = 8$ (XXZ, LiH) and $q \approx 288$ for $n = 12$ (H_2O). Each iteration generates $O(n)$ perturbed configurations, which are inserted into a ranked list with $O(\log q)$ cost (or $O(1)$ using a heap implementation).The weighted LSQ step solves for $O(n^2)$ parameters corresponding to the constant, linear, and pairwise couplings $(c_0, \vec{l}, \vec{J})$ and is dominated by matrix multiplications, rendering its cost negligible compared to RBM training and Monte Carlo sampling. The subsequent BFGS refinement optimizes the same $O(n^2)$ parameters and empirically converges within 20–50 iterations across all problem instances, leading to surrogate-update times of order 10^{-2}–10^{-1} seconds for $n = 8$–12. Consequently, the surrogate-fitting overhead contributes only a few percent of the total wall-clock time per training iteration.

Stability analysis: The two-stage LSQ + BFGS protocol was found to be highly robust across all benchmark systems. The weighted LSQ step employs weights $T_{ii} = \sqrt{\rho_{\upsilon}^{\upsilon}(\upsilon^{(i)})}$, suppressing the influence of configurations with exponentially small probabilities whose logarithms would otherwise dominate the fit. This reweighting emphasizes statistically relevant high-probability regions while retaining sufficient coverage through additional random samples. Furthermore the resulting LSQ problem remains strictly convex in $\vec{\theta} = (c_0, \vec{l}, \vec{J})$ due to nature of the cost-function and thus admits a unique global minimum. From a linear-algebra perspective, the weighting improves conditioning by rescaling rows of the design matrix according to statistical relevance. In the unweighted formulation, rows associated with exponentially small $\rho_{\upsilon}^{\upsilon}(\upsilon)$ produce large residuals and highly disparate row norms, leading to an ill-conditioned matrix $A^T A$. The weighted variant $(TA)^T(TA)$ removes this problem and exhibits significantly reduced condition number in practice, yielding stable inversions and reliable parameter estimates. Empirically, convex LSQ optimization provides stable initial parameters near the desired the minima to warm-start the subsequent BFGS step. This step is important as it further reduces residual error always by one to two orders of magnitude with smooth monotonic convergence due to the low dimensionality of the surrogate parameter space, and the smooth, well-conditioned cost function employed.

In summary, for the system sizes and training regimes explored in this work, the surrogate-fitting stage scales polynomially in n, uses $O(n^2)$ samples, incurs negligible runtime overhead relative to RBM training and Monte Carlo sampling, and is numerically stable across diverse physical problem instances by a two-pronged design of LSQ+BFGS. Furthermore we must emphasize that this algorithmic step wont be necessary at all if we use an universal phase functional embedded into the very choice of the NQS, like what we designed for the amplitude functional in this work. This can be analyzed in a further work.

S4. COMPARISONS WITH MISCELLANEOUS ADAPTIVE CLASSICAL PROPOSAL DISTRIBUTIONS

In Fig.3(b-c) we showed comparisons with a handful of candidate proposal distributions commonly used for generating prospective candidates for $\vec{\upsilon}$. Apart from local proposal (A), the other two classical proposals used were Uniform (B) and Haar-random (C). The latter two were global updates as they can potentially mutate the incumbent configuration at all sites. However none of them were adaptive i.e. used information about the target distribution. To ensure an exhaustive comparison we present in this section a thorough comparison with a suite of advanced adaptive classical proposal distributions with quantum-enhanced proposals. To quantify the comparisons, lets recapitulate the setup we have used for Markov Chain Monte Carlo. Given access to an instantaneous configuration (say $\vec{\upsilon}^{(i)} \in \{1, -1\}^n$) to draw samples from the target distribution $\phi(\vec{\upsilon})$ (see main text) we are

essentially drawing samples from a transition matrix with elements $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ defined as

$$\begin{aligned} & T(\vec{v}^{(i+1)}|\vec{v}^{(i)}) \\ & = min\left(1, \frac{\phi(\vec{v}^{(i+1)})P_{prop}(\vec{v}^{(i)}|\vec{v}^{(i+1)})}{\phi(\vec{v}^{(i)})P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)})}\right) \\ & \times P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)}) \end{aligned} \tag{S10}$$

The definition above in Eq.S10 for $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ enforces detailed balance with $\phi(\vec{v})$ and hence guarantees convergence in the long-time limit to the the steady distribution $\phi(\vec{v})$ which is also the target. To initiate the Markov chain, the user will have to first draw samples from $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ which defines the transition matrix elements in Eq.S10 above. Different proposal distributions $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ can accelerate or deccelerate the convergence time This is best captured in a quantity called the mixing time[1] (t_{mix}), which roughly estimates the minimum number of samples drawn before the Markov chain converges to the target distribution with an error threshold (say e) in the total variation distance. The following identity (also see main text) provides a robust upper and lower bound to the mixing time as follows[2]:

$$\left(\delta^{-1}-1\right)\ln\left(\frac{1}{2e}\right) \leq t_{mix} \leq \delta^{-1}\ln\left(\frac{1}{e\,\min_{\vec{v}}\phi(\vec{v})}\right) \tag{S11}$$

where δ is the absolute spectral gap [3, 4] for the transition matrix $\vec{T}$ (see Eq.S10). Formally $\delta = min\{\lambda_0 - \lambda_i\}_{i=1} \in \mathbb{R}_+$ where as defined in the text λ_i is the ith eigenvalue of $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ in a sorted list with $\lambda_i > \lambda_{i-1}\ \forall i$.

The comparison is made concrete by computing the absolute spectral gap δ choosing 250 randomly chosen instances of $(J(\vec{X}), l(\vec{X})$ to extract typical average case behavior and its spread (quantified by standard deviations as error bars in the plots). We also do the comparison for various sizes across $n = 3, 4, 5, 6, 7, 8, 9, 10$ to show how the spectral gap scales as a function of system size in a similar fashion as done in Fig.3(b) for certain set of non-adaptive classical proposals. Since we defined a vast array of quantum-enabled proposals (D-H) but used the Trotterized version (Proposal H in main text) for all data-generation due to ease of implementation on the quantum circuit, we shall focus on comparisons of particularly Proposal H and Proposal D (average quantum case when averaged over many instances of τ and γ) with other adaptive proposal distributions in this section. The suite of advanced classical proposals $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ used can be sub-categorized as follows:

1. Classical local update proposal - This corresponds to Proposal A in the manuscript and has already been included in page 6 and Fig.3(b-c). We shall replicate the absolute spectral gap vs number of spins (n) for the transition matrix associated with this proposal (see Eq.S10) and the quantum variants (specifically Proposal D and Proposal H) below separately in Fig.S2(a). The conclusions are exactly what was discussed in main text, the spectral gap of the quantum ones are higher (indicating lower mixing time) and decreases at a cubically slower rate (see inset for slope) when system size is increased.

2. Classical cluster update proposals - These proposal distributions corresponds to choosing a subset of spin sites (which is called a cluster) from the current configuration $\vec{v}^{(i)}$ and involves flipping/exchanging the spins within this cluster to generate new configuration $\vec{v}^{(i+1)}$. We have considered two sub-categories of this proposal type

   - **Non-adaptive cluster update proposals**
     - i **Cluster update by pair-exchange** - This corresponds to choosing two spin-sites (say (k, m)) at random from the present configuration $\vec{v}^{(i)}$ and exchanging the components at the two sites i.e. setting $\vec{v}_k^{(i+1)} = \vec{v}_m^{(i)}$ and $\vec{v}_m^{(i+1)} = \vec{v}_k^{(i)}$. The spin components at remaining sites remain unchanged. One must note that due to pair-exchange, this proposal distribution generates configurations with conserved Hamming weights and can be strongly non-ergodic.
     - ii **Cluster update by co-ordinated pair flip** - This corresponds to choosing two spin-sites (say (k, m)) at random from the present configuration $\vec{v}^{(i)}$ and flipping the components at the chosen sites (not exchanging them like above). The spin components at remaining sites remain unchanged. This generates a new configuration $\vec{v}^{(i+1)}$ on which the procedure is repeated.
   - **Adaptive cluster update proposals**
     - i **Swendsen-Wang updates (Wolff type clustering)** - This corresponds to introducing a bond between all pairs of parallel spins (say a representative pair is (k, m)) in the current configuration $\vec{v}^{(i)}$ with a probability given by $1 - e^{-2\beta J_{km}}$ where J_{km} is the strength of the interaction between the spin pair (k, m) as defined in the text. This is the adaptive step as explicit information about the target distribution $\phi(\vec{v}) \propto e^{-\beta(\sum_i l_i(\vec{X})v_i + \sum_{i,j} J_{ij}v_iv_j)}$ is utilized. In the original Swendsen-Wang[5] many such clusters are updated at once whereas in the Wolff modification , a single such cluster is grown and is then subsequently flipped in each update to spawn a new configuration $\vec{v}^{(i+1)}$. Further details about this procedure can be found in [6–9].

For all the sub-categories of cluster based updates, we plot the absolute spectral gap $|\delta|$ vs number of spins (n) and compare it with quantum-enhanced proposals D, H in Fig.S2(b) and see that the spectral gaps for these cluster updates are lower and decreases nearly cubically faster (see inset for slope) with N than quantum ones. This formally indicates larger mixing time in classical proposals of this family compared to the quantum ones.

3. Classical global update proposals

- **Non-adaptive global update proposals**

  i **Uniform updates** - This corresponds to choosing any configuration from the present one $\vec{v}^{(i)}$ with equal probability. This means the proposal is simply $P(\vec{v}^{(i+1)}|\vec{v}^{(i)}) = \frac{1}{2^n}$. This was already included as Proposal B in pages 6-7 and Fig.3(b-c) in main manuscript. We shall replicate the absolute spectral gap vs number of spins (n) for the transition matrix associated with this proposal (see Eq.S10) and the quantum variants (specifically Proposal D and Proposal H) below separately in Fig.S3(a). The conclusions are exactly what was discussed in main text, the spectral gap of the quantum ones are higher (indicating lower mixing time) and decreases at a cubically slower rate (see inset for slope) when system size is increased.

  ii **Haar-random update** - This corresponds to choosing a proposal distribution of the kind $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)}) = |\langle\vec{v}^{(i+1)}|U|\vec{v}^{(i)}\rangle|^2$ where $U \sim P_{Haar}(U)$ i.e sampling a random unitary from the Haar measure. [10, 11]. This was included in the too as Proposal C in pages 6-7 and Fig.3(b-c) of main manuscript. We shall replicate the absolute spectral gap vs number of spins (n) for the transition matrix associated with this proposal (see Eq.S10) and the quantum variants (specifically Proposal D and Proposal H) below separately in Fig.S3(a). The conclusions are exactly what was discussed in main text, the spectral gap of the quantum ones are higher (indicating lower mixing time) and decreases at a cubically slower rate (see inset for slope) when system size is increased. Furthermore we see more clearly than Proposal C and Proposal B yields transition matrices with near identical spectral gaps. This is not a coincidence as we shall formally prove in Section **??** that on an average $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ generated from Haar random unitaries behaves like the uniform case.

- **Adaptive global update proposals**

  i **Random-walk Metropolis (RWM) algorithm based update with a preconditioner (P)** - This proposal [12–14] requires embedding the discrete variable $\vec{v}^{(i)}$ within a continuous function as $\vec{v}^{(i)} = \text{sgn}(\tanh(\alpha\vec{x}^{(i)}))$ and then updating the new configuration through the following proposal

$$\begin{aligned}
\vec{x}^{(i+1)} &= \vec{x}^{(i)} - \epsilon \mathrm{P}\vec{\eta} \\
\vec{\eta} &\sim \mathcal{N}(0, \mathbb{I}) \\
\mathrm{P} &= (\mathrm{Cov}(\vec{x}_{\leq i}) + \lambda\mathbb{I})^{\frac{1}{2}} \\
\mathrm{Cov}(\vec{x}_{\leq i}) &= \mathbb{E}[(\vec{x}_{\leq i} - \mathbb{E}[\vec{x}_{\leq i}])(\vec{x}_{\leq i} - \mathbb{E}[\vec{x}_{\leq i}])^T] \sim \sum_{j<i}^{N_j}(\vec{x}^{(j)} - \frac{\sum_{j<i}^{N_j}\vec{x}^{(j)}}{N_j})(\vec{x}^{(j)} - \frac{\sum_{j<i}^{N_j}\vec{x}^{(j)}}{N_j})^T \\
\epsilon,\ \lambda &\to 0_+ \in \mathbb{R}
\end{aligned} \tag{S12}$$

  The sampled noise variable η is gaussian and P serves as the preconditioner and is the Cholesky decomposition of the regularized Covariance matrix ($\mathrm{Cov}(\vec{x}_{\leq i})$) which is empirically estimated using a subset of past drawn samples (N_j being the size of the subset defined by the user). This makes the update rule adaptive as it depends on past accepted samples. The new updated continuous variable $\vec{x}^{(i+1)}$ is reconverted to discrete variable $\vec{v}^{(i+1)}$ through the inverse embedding map. Collection of statistics from many independent runs starting from the same $\vec{x}^{(i)}$ can allow one to construct the matrix $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Each element of this matrix is then multiplied with the pre-factor governing acceptance criteria in Eq.S10 to generate $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Alternatively, due to the gaussianity of the drawn noise field, the above update is formally equivalent to a proposal as

$$\vec{x}^{(i+1)} \sim P_{prop}(\vec{x}^{(i+1)}|\vec{x}^{(i)}) = \mathcal{N}(\vec{x}^{(i)}, \epsilon^2\mathrm{Cov}(\vec{x}_{\leq i})) \tag{S13}$$

  ii **Metropolis adjusted Langevin algorithm (MALA) based update with a preconditioner (P)** - This proposal [12–17] is similar RWM and requires embedding the discrete variable $\vec{v}^{(i)}$ within a continuous function as $\vec{v}^{(i)} = \text{sgn}(\tanh(\alpha\vec{x}^{(i)}))$. The only difference lies in the use of the energy gradient $\vec{\nabla}(E(\vec{x}))_{\vec{x}=\vec{x}^{(i)}}$ which is used to update the current configuration as follows

$$\vec{x}^{(i+1)} = \vec{x}^{(i)} - \frac{\epsilon^2}{2}\vec{\nabla}(E(\vec{x}))_{\vec{x}=\vec{x}^{(i)}}\mathrm{Cov}(\vec{x}_{\leq i}) - \epsilon\mathrm{P}\vec{\eta} \tag{S14}$$

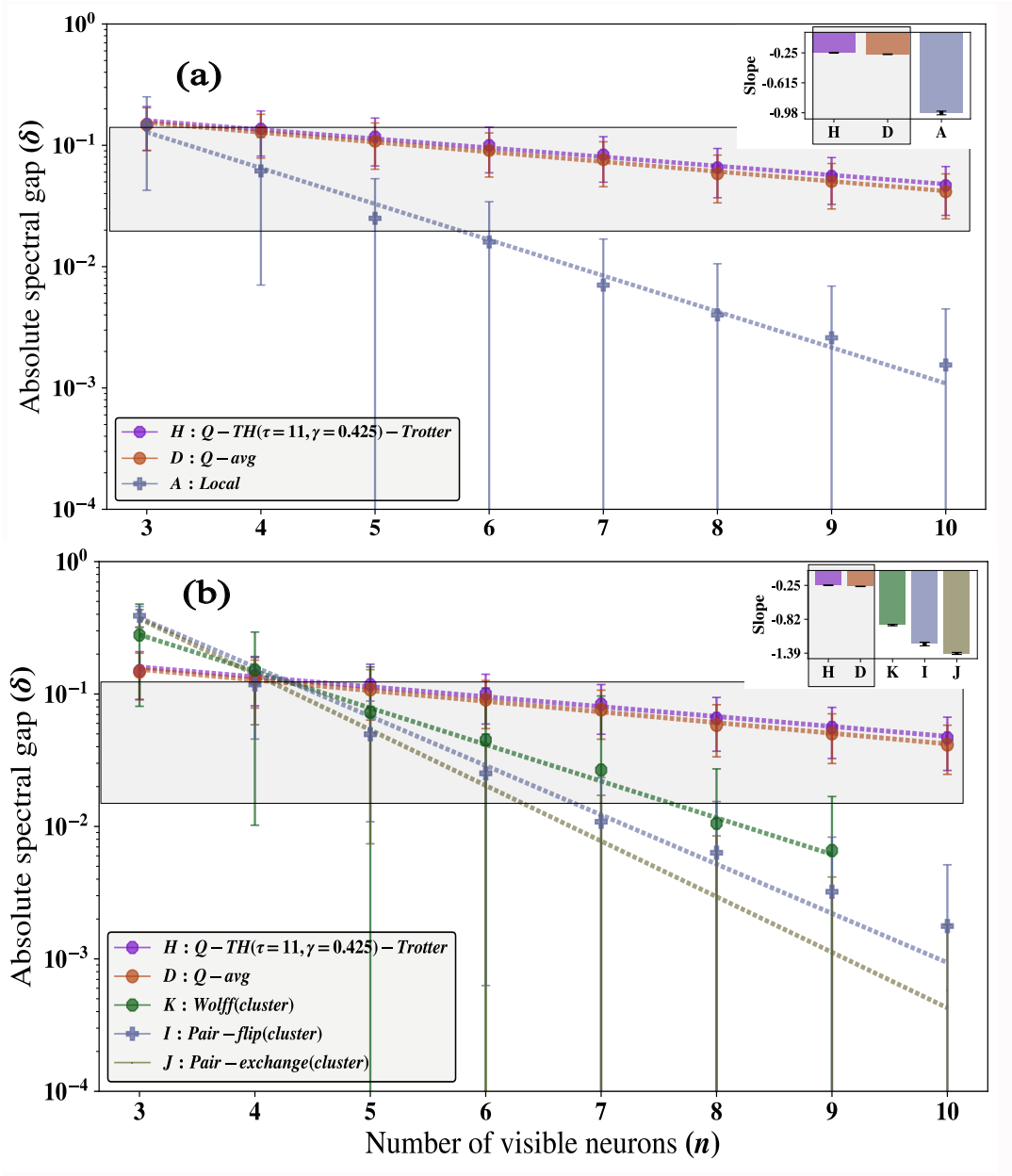

FIG. S2. **Comparison with cluster and local updates**(a) The absolute spectral gap δ as a function of number of visible neurons/system size (n) for the transition matrix $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ generated from Proposal A (Local update) in blue and quantum-enhanced proposals (Proposal D and Proposal H in main text, see pages 6-7) in orange and purple respectively. We see that not only the quantum-enhanced proposals have higher spectral gap δ (lower mixing time, see Eq.S11) but the gap also decreases at a slower rate with n by almost factor of 3-4 compared to Proposal A (see inset at the top-right for the respective slopes). This means the quantum proposals D,H will faster convergence to the steady distribution. The curves for Proposals D,H (as well the slopes in the inset) are highlighted within a gray box as a guide to the eye. (b) Same as in (a) but for comparing proposal distributions with cluster update schemes i.e. (i) Cluster update by co-ordinated pair flip (denoted as Proposal I/Pair-flip(cluster)) in blue (ii) Cluster update by co-ordinated pair exchange (denoted as Proposal I/Pair-exchange(cluster)) in olive and (iii) Swendsen-Wang updates/Wolff type clustering denoted as Proposal K (Wolff(cluster)) in dark green vs the two quantum-enhanced proposals (D,H) in orange and purple respectively. We see except at n=3 (at n=4 there is a crossover) , for all system sizes the quantum-enhanced proposals have higher spectral gap δ (lower mixing time, see Eq.S11). Also the δ (gap) values are decreasing at nearly 4.5 times faster rate for Proposal I-J and about 3 times for Proposal K (see slope plot in the inset at top-right) than the quantum-enhanced ones (D,H). The curves for Proposals D,H (as well the slopes in the inset) are highlighted within a gray box as a guide to the eye. For Proposal K (Wolff) in dark green, the point corresponding to $n = 10$ requires a memory intensive computation for the full transition matrix and hence has been omitted. But the trendline is clear. For each n in both a-b, we have used 250 random instances of $(\vec{l}(\vec{X}), \vec{J}(\vec{X}))$ (similar to that in Fig.3(b-c)) to display average behavior. Fluctuations from this average is quantified using standard deviation as error bars.

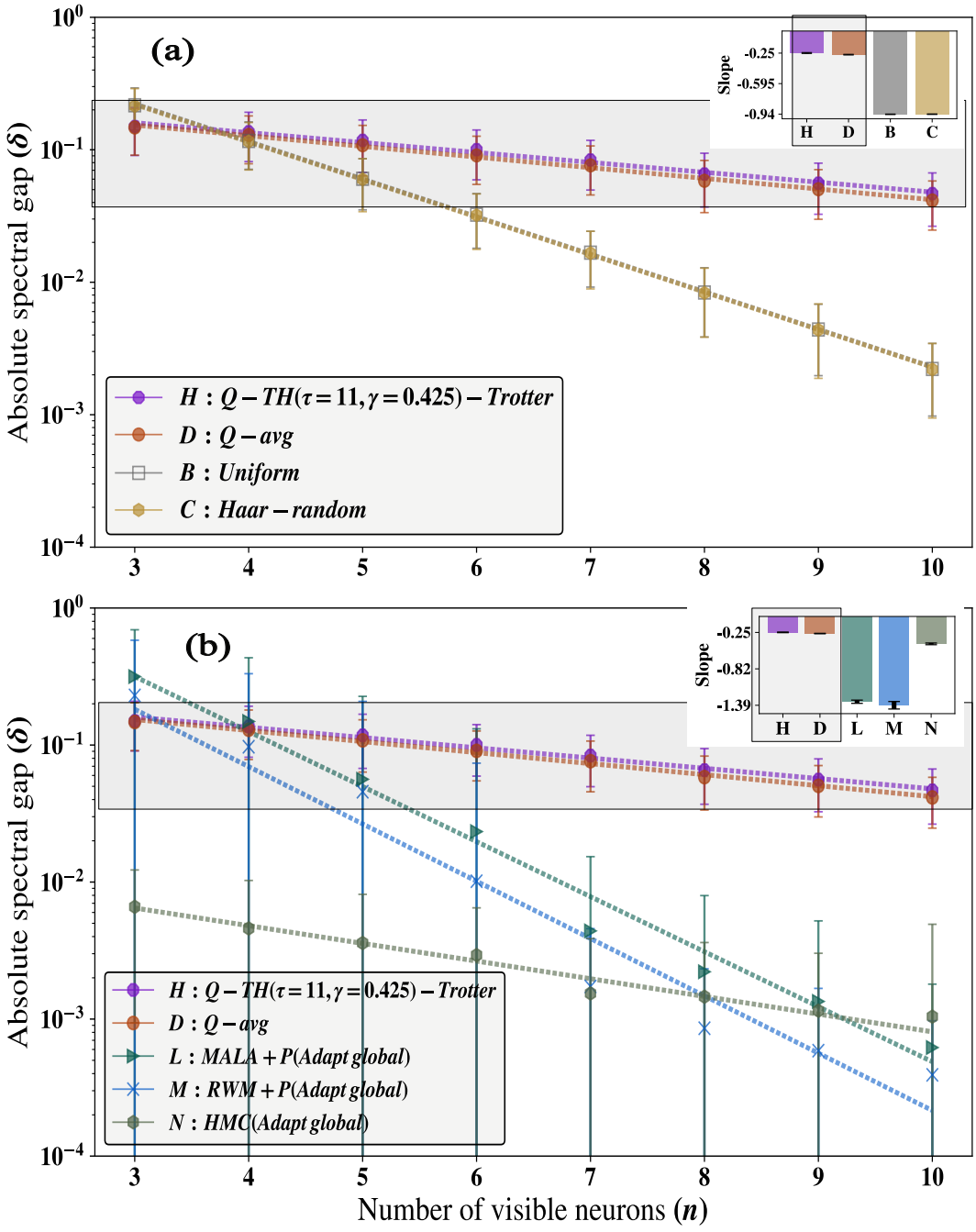

FIG. S3. **Comparison with adaptive and non-global updates**(a) The absolute spectral gap δ as a function of number of visible neurons/system size (n) for the transition matrix $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ generated from non-adaptive global update schemes i.e. Proposal B (Uniform proposal update) in gray, Proposal C (Haar random unitary update) in yellow and quantum-enhanced proposals (Proposal D and Proposal H in main text, see pages 6-7) in orange and purple respectively. We see that not only the quantum-enhanced proposals have higher spectral gap δ (lower mixing time, see Eq.S11) for all n beyond 4 ($n = 4$ is a crossover point) but the gap also decreases at a slower rate with n by almost factor of 3.5 compared to Proposal B and C (see inset at the top-right for the respective slopes). This means the quantum proposals D,H will faster convergence to the steady distribution. The curves for Proposals D,H (as well the slopes in the inset) are highlighted within a gray box as a guide to the eye. Furthermore the fact that the spectral gaps from Proposal B and C are identical is not a coincidence. We rigorously prove in Supplementary Information **??** that Proposal C on an average will yield Proposal B always and any instance-based fluctuations about that average behavior is exponentially suppressed in system size n. (b) Same as in (a) but for comparing proposal distributions with adaptive global update schemes i.e. (i) MALA + P (denoted as Proposal L) in dark green (ii) RWM + P (denoted as Proposal M) in light blue and (iii) HMC (denoted as Proposal N) in olive vs the two quantum-enhanced proposals (D,H) in orange and purple respectively. We see except at n=3 (at n=4 there is a crossover) , for all system sizes the quantum-enhanced proposals have higher spectral gap δ (lower mixing time, see Eq.S11). Also the δ (gap) values are decreasing at nearly 4.5 times faster rate for Proposal L-M and about 1.75 times for Proposal N (see slope plot in the inset at top-right) than the quantum-enhanced ones (D,H). The curves for Proposals D,H (as well the slopes in the inset) are highlighted within a gray box as a guide to the eye. For each n in both a-b, we have used 250 random instances of $(\vec{l}(\vec{X}), \vec{J}(\vec{X}))$ (similar to that in Fig.3(b-c)) to display average behavior. Fluctuations from this average is quantified using standard deviation as error bars. For HMC 30,000 $\vec{p}$ samples were used for each $\vec{x}$ (see text) which is 3 times larger than the number of samples used for quantum-enhanced proposals.

The energy function $E(\vec{x}) = -\frac{1}{\beta}\log(\phi(\vec{v} = \text{sgn}(\tanh(\alpha\vec{x}))))$ is defined with $\phi(\vec{v})$ being the target distribution $\phi(\vec{v}) \propto e^{-\beta(\sum_i l_i(\vec{X})v_i + \sum_{i,j} J_{ij}v_iv_j)}$ as mentioned in the main text. This thereby makes the proposal adaptive beyond the usage of estimated covariances as in RWM. For numerical evaluation of gradients we use the embedding $\alpha \to \infty$ and drop the $sgn(.)$. The meaning of all other symbols $(\text{P}, \eta, \lambda, \epsilon)$ remain the same as defined above for RWM. P as before is the Cholesky decomposed empirically estimated regularized Covariance matrix $(\text{Cov}(\vec{x}_{\leq i}))$ from past samples. The new updated continuous variable $\vec{x}^{(i+1)}$ is reconverted to discrete variable $\vec{v}^{(i+1)}$ through the inverse embedding map defined above. Collection of statistics from many independent runs starting from the same $\vec{x}^{(i)}$ can allow one to construct the matrix $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Each element of this matrix is then multiplied with the pre-factor governing acceptance criteria in Eq.S10 to generate $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Alternatively, due to the gaussianity of the drawn noise field, the above update is formally equivalent to sampling from a gaussian with displaced mean as

$$\vec{x}^{(i+1)} \sim P_{prop}(\vec{x}^{(i+1)}|\vec{x}^{(i)}) = \mathcal{N}(\vec{x}^{(i)} - \frac{\epsilon^2}{2}\vec{\nabla}(E(\vec{x}))_{\vec{x}=\vec{x}^{(i)}}\text{Cov}(\vec{x}_{\leq i}), \epsilon^2\text{Cov}(\vec{x}_{\leq i})) \tag{S15}$$

iii **Hamiltonian Monte Carlo (HMC) with auxillary momentum field** - This proposal is similar RWM and MALA and requires embedding the discrete variable $\vec{v}^{(i)}$ within a continuous function as $\vec{v}^{(i)} = \text{sgn}(\tanh(\alpha\vec{x}^{(i)}))$. However unlike the above two, it also assumes access to an auxillary momentum field $\vec{p}$ to define an overall classical Hamiltonian[14, 18–21]. This classical Hamiltonian is defined as

$$\begin{aligned} H(\vec{x}, \vec{p}) &= \frac{\vec{p}.\vec{p}}{2} + E(\vec{x}) \\ \vec{p} &\sim \mathcal{N}(0, \mathbb{I}) \\ E(\vec{x}) &= -\frac{1}{\beta}\log(\phi(\vec{v} = \text{sgn}(\tanh(\alpha\vec{x})))) \end{aligned} \tag{S16}$$

where $\phi(\vec{v}) \propto e^{-\beta(\sum_i l_i(\vec{X})v_i + \sum_{i,j} J_{ij}v_iv_j)}$ is the target distribution. For a given choice of $\vec{p} \sim \mathcal{N}(0, \mathbb{I})$ and with the current configuration $\vec{v}^{(i)}$, one defines an initial phase-space configuration $(\vec{x}^{(i)}, \vec{p})$ which is propagated to a final set of phase-space co-ordinates $(\vec{x}^{(i+1)}, \vec{p}^*)$ by integrating Hamilton's equation of motion using leapfrog integrator. This can be understood as follows:

$$\begin{aligned} \frac{\partial\vec{x}}{\partial t} &= \frac{\partial H(\vec{x}, \vec{p})}{\partial\vec{p}} = \vec{p} \\ \frac{\partial\vec{p}}{\partial t} &= -\frac{\partial H(\vec{x}, \vec{p})}{\partial\vec{x}} = -\vec{\nabla}E(\vec{x}) \end{aligned}$$

Leapfrog integrator is implemented for L steps to solve the above coupled set of first-order differential equations as follows:

$$\vec{p}_{t+0.5\epsilon} = \vec{p}_t + \frac{\epsilon}{2}\frac{\partial\vec{p}}{\partial t}|_{\vec{x}_t} \tag{S17}$$

$$\begin{aligned} \vec{x}_{t+\epsilon} &= \vec{x}_t + \epsilon\vec{p}_{t+0.5\epsilon} \\ \vec{p}_{t+\epsilon} &= \vec{p}_{t+0.5\epsilon} + \frac{\epsilon}{2}\frac{\partial\vec{p}}{\partial t}|_{\vec{x}_{t+\epsilon}} \end{aligned} \tag{S18}$$

For numerical evaluation of gradients we use the embedding $\alpha \to \infty$ and drop the $sgn(.)$. After these L steps, the final set of co-ordinates so obtained is $(\vec{x}^{(i+1)}, \vec{p}^*)$ which can be denoted as a transformation $(\vec{x}^{(i+1)}, \vec{p}^*) = \mathcal{L}[(\vec{x}^{(i)}, \vec{p})]$ with $\mathcal{L}$ being the leapfrog integrator. Projecting this onto co-ordinate space alone followed by integrating over many initial choices of $\vec{p}$ gives us the continuous-space embedding of the proposal distribution

$$P_{prop}(\vec{x}^{(i+1)}|\vec{x}^{(i)}) = \int \mathcal{N}_{\vec{p}}(0, \mathcal{I})\delta(\vec{x}^{(i+1)} - \Pi_{\vec{x}}\mathcal{L}[(\vec{x}^{(i)}, \vec{p})])d\vec{p} \tag{S19}$$

with $\Pi_{\vec{x}}$ being projection of $(\vec{x}^{(i+1)}, \vec{p}^*)$ into co-ordinate space. Given $\vec{x}^{(i+1)} \sim P_{prop}(\vec{x}^{(i+1)}|\vec{x}^{(i)})$, the new updated continuous variable $\vec{x}^{(i+1)}$ is reconverted to discrete variable $\vec{v}^{(i+1)}$ through the inverse embedding map defined above. Collection of statistics from many independent runs starting from the same $\vec{x}^{(i)}$ can allow one to construct the matrix $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Each element of this matrix is then multiplied with the pre-factor governing acceptance criteria in Eq.S10 to generate $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$.

For all the sub-categories of adaptive global update proposals, we again plot the absolute spectral gap $|\delta|$ vs number of spins (n) and compare it with quantum-enhanced proposals D, H in Fig.S3(b). For RWM+P and MALA+P, we see that spectral gaps even though higher at n=3, the gap decreases fast nearly quartically as a function of system size compared to quantum ones (D, H). For HMC we see that the decrease in spectral gap decreases slower than MALA+P, RWM+P which is much better but nonetheless still the rate of decrease is faster by a factor of 1.75 compared to the quantum ones (D,H). Also the spectral gaps across the entire range are consistently lower than the quantum ones. The fact that HMC has mixing time which is better than MALA +P and RWM+P has been investigated in recent times [22] and is largely a consequence of the fact that MALA can be recasted as HMC with a single leap-frog step whereas in conventional HMC as has been used here the number of leapfrog steps for integration is higher than 1 (we have 10 steps per momentum $\vec{p}$ sample) . It must be mentioned that 30,000 $\vec{p}$ samples are drawn for performing the integral in Eq.S19 which is larger than the number of samples drawn from the quantum proposal H in generating all data in Fig.3, Fig.5, Fig.6 and Fig.7. Also for each p, leapfrog integration is necessary to update the Hamilton's equations. This makes generation of the transition matrix for each n much slower than for the quantum case where no integration over an auxillary variable is necessary. On top of that , the results in Fig.S3(b) formally indicates larger mixing time in all classical proposals of this family compared to the quantum ones.

S5. GRADIENT EXPRESSIONS

In this section, we derive the gradient expressions for the state ansatz with respect to its parameters. These gradients are crucial for training the model parameters. We provide analytical forms for the gradient of the density matrix and the expectation value of observables.

Lemma 4. *$\partial_{x_i}\rho^{v}_{v'} = \mathcal{D}^{v}_{v'}(x_i) \odot \rho^{v}_{v'}$; where the '$\odot$' represents element-wise Hadamard product between matrices and the matrix $\mathcal{D}^{v}_{v'}(x_i)$ for various parameters x_i is given as follows:*

x_i	$\mathcal{D}^{v}_{v'_i}(x_i)$
$\mathrm{Re}(a_k)$	$-\beta\left(v_k + v'_k\right)$
$\mathrm{Im}(a_k)$	$-i\beta\left(v_k - v'_k\right)$
$\mathrm{Re}(b_p)$	$\beta\Big\{ \tanh\left(\beta b_p + \beta\sum_{i=1}^{n} W_{ip}v_i\right)$ $+\tanh\left(\beta b^*_p + \beta\sum_{i=1}^{n} W^*_{ip}v'_i\right)\Big\}$
$\mathrm{Im}(b_p)$	$i\beta\Big\{ \tanh\left(\beta b_p + \beta\sum_{i=1}^{n} W_{ip}v_i\right)$ $-\tanh\left(\beta b^*_p + \beta\sum_{i=1}^{n} W^*_{ip}v'_i\right)\Big\}$
$\mathrm{Re}(W_{kp})$	$\beta\Big\{ \tanh\left(\beta b_p + \beta\sum_{i=1}^{n} W_{ij}v_i\right)v_k$ $+\tanh\left(\beta b^*_p + \beta\sum_{i=1}^{n} W^*_{ip}v'_i\right)v'_k\Big\}$
$\mathrm{Im}(W_{kp})$	$i\beta\Big\{ \tanh\left(\beta b_p + \beta\sum_{i=1}^{n} W_{ip}v_i\right)v_k$ $-\tanh\left(\beta b^*_p + \beta\sum_{i=1}^{n} W^*_{ip}v'_i\right)v'_k\Big\}$

Proof. The goal is to compute the derivative of $\rho^{v}_{v'}$ with respect to various parameters x_i. The state $\rho(\vec{x})$ is defined as:

$$\rho^{v}_{v'}(\vec{x}) \propto exp\left(-\beta\sum_i a_i v_i + \sum_i a^*_i v'_i\right)\prod_{j=1}^{m}\Gamma_j(\vec{v},\vec{b},\vec{W})\,\Gamma_j(\vec{v}',\vec{b}^*,\vec{W}^*)$$

where, $\Gamma_j(\vec{v},\vec{b},\vec{W}) = \cosh\left[\beta\left(b_j + \sum_{i=1}^{n} W_{ij}v_i\right)\right]$
We take the logarithmic derivative of $\rho^{v}_{v'}$ with respect to x_i:

$$\boxed{\partial_{x_i}\rho^{v}_{v'} = \rho^{v}_{v'}\cdot\partial_{x_i}\ln\rho^{v}_{v'}}$$

We denote $\mathcal{D}^{v}_{v'}(x_i) = \partial_{x_i}\ln\rho^{v}_{v'}$. Therefore $\partial_{x_i}\rho^{v}_{v'} = \mathcal{D}^{v}_{v'}(x_i)\odot\rho^{v}_{v'}$; where the '$\odot$' represents element-wise Hadamard product between matrices and the matrix $\mathcal{D}^{v}_{v'}(x_i)$
The logarithm of $\rho^{v}_{v'}$ is:

$$\ln\rho^{v}_{v'} = -\beta\sum_i\left(a_i v_i + a^*_i v'_i\right) + \sum_{j=1}^{m}\left[\ln\Gamma_j(\vec{v},\vec{b},\vec{W}) + \ln\Gamma_j(\vec{v}',\vec{b}^*,\vec{W}^*)\right]$$

Now, we take the derivative of each term w.r.t the parameters x_i:
- For the first term:

$$\partial_{x_i}\left[-\beta\sum_i\left(a_i v_i + a^*_i v'_i\right)\right]$$

is nonzero only if x_i corresponds to $\mathrm{Re}(a_k)$ or $\mathrm{Im}(a_k)$, giving the contributions:

$$\boxed{\partial_{\mathrm{Re}(a_k)} = -\beta(v_k + v'_k)} \qquad \boxed{\partial_{\mathrm{Im}(a_k)} = -i\beta(v_k - v'_k)}$$

- For the second term,

$$\ln \Gamma_j = \ln \cosh \left(\beta b_j + \beta \sum_{i=1}^{n} W_{ij} v_i \right),$$

depends only b and W parameters.

- For b_p:

  Using $\tanh x = \partial_x (\ln \cosh x)$, we have:

$$\partial_{\mathrm{Re}(b_p)} \ln \Gamma_j = \tanh \left(\beta b_p + \beta \sum_{i=1}^{n} W_{ip} v_i \right).$$

  Similarly, for the conjugate term from $\Gamma_j(\vec{v}', \vec{b}^*, \vec{W}^*)$:

$$\partial_{\mathrm{Re}(b_p)} \ln \Gamma'_j = \tanh \left(\beta b_p^* + \beta \sum_{i=1}^{n} W_{ip}^* v'_i \right).$$

  Adding these:

$$\boxed{\partial_{\mathrm{Re}(b_p)} \ln \rho^v_{v'} = \beta \left[\tanh \left(\beta b_p + \beta \sum_{i=1}^{n} W_{ip} v_i \right) + \tanh \left(\beta b_p^* + \beta \sum_{i=1}^{n} W_{ip}^* v'_i \right) \right].}$$

  Similarly for $\mathrm{Im}(b_p)$,

$$\boxed{\partial_{\mathrm{Im}(b_p)} \ln \rho^v_{v'} = i\beta \left[\tanh \left(\beta b_p + \beta \sum_{i=1}^{n} W_{ip} v_i \right) - \tanh \left(\beta b_p^* + \beta \sum_{i=1}^{n} W_{ip}^* v'_i \right) \right].}$$

- For W_{kp}:

  The derivative includes the dependence on v_k and v'_k:

$$\boxed{\partial_{\mathrm{Re}(W_{kp})} \ln \rho^v_{v'} = \beta \left[\tanh \left(\beta b_p + \beta \sum_{i=1}^{n} W_{ip} v_i \right) v_k + \tanh \left(\beta b_p^* + \beta \sum_{i=1}^{n} W_{ip}^* v'_i \right) v'_k \right],}$$

  and similarly for $\mathrm{Im}(W_{kp})$:

$$\boxed{\partial_{\mathrm{Im}(W_{kp})} \ln \rho^v_{v'} = i\beta \left[\tanh \left(\beta b_p + \beta \sum_{i=1}^{n} W_{ip} v_i \right) v_k - \tanh \left(\beta b_p^* + \beta \sum_{i=1}^{n} W_{ip}^* v'_i \right) v'_k \right].}$$

□

Lemma 5. $\partial_{x_i} \langle O \rangle (\vec{X}) = \left\langle \left[\mathcal{D}_{x_i} \odot O^T \right]_\rho (v) \right\rangle_{\rho^v_v} - \left\langle \mathcal{D}_{x_i}(v) \right\rangle_{\rho^v_v} \left\langle O_\rho(v) \right\rangle_{\rho^v_v}$

Proof.

$$
\begin{aligned}
\langle O \rangle(\vec{X}) &= \frac{Tr(O\rho(\vec{X}))}{Tr(\rho(\vec{X}))} \\
\partial_{x_i}\langle O \rangle(\vec{X}) &= \partial_{x_i}\left(\frac{Tr(O\rho(\vec{X}))}{Tr(\rho(\vec{X}))}\right) \\
&= \partial_{x_i}\left(\frac{\sum_{v,v'} \rho_{v'}^{v} O_{v}^{v'}}{\sum_v \rho_v^v(\vec{X})}\right) \\
&= \frac{\left(\sum_{v,v'} \partial_{x_i}\rho_{v'}^{v} O_{v}^{v'}\right)}{\sum_v \rho_v^v(\vec{X})} - \frac{\left(\sum_v \partial_{x_i}\rho_v^v\right)\left(\sum_{v,v'} \rho_{v'}^{v} O_{v}^{v'}\right)}{\left(\sum_v \rho_v^v(\vec{X})\right)^2} \\
&= \frac{\sum_{v,v'} \mathcal{D}_{v'}^{v}(x_i)\rho_{v'}^{v} O_{v}^{v'}}{\sum_v \rho_v^v(\vec{X})} - \left(\frac{\sum_v \mathcal{D}_v^v(x_i)\rho_v^v}{\sum_v \rho_v^v(\vec{X})}\right)\left(\frac{\sum_{v,v'} \rho_{v'}^{v} O_{v}^{v'}}{\sum_v \rho_v^v(\vec{X})}\right) \\
&= \sum_v \rho_v^v \left(\frac{\sum_{v'} \rho_{v'}^{v}\left[\mathcal{D}_{x_i} \odot O^T\right]_v^{v'}}{\rho_v^v}\right) \Big/ \sum_v \rho_v^v(\vec{X}) - \left\langle \mathcal{D}_{x_i}(v) \right\rangle_{\rho_v^v} \left\langle O_\rho(v) \right\rangle_{\rho_v^v} \\
&= \left\langle \left[\mathcal{D}_{x_i} \odot O^T\right]_\rho (v) \right\rangle_{\rho_v^v} - \left\langle \mathcal{D}_{x_i}(v) \right\rangle_{\rho_v^v} \left\langle O_\rho(v) \right\rangle_{\rho_v^v}
\end{aligned}
$$

By substituting $O = H$, i.e. the hamiltonian of the driver system in the above expression, we get analytical gradients used for training parameters $\vec{X}$. □

S6. QUANTUM CIRCUITS

The sampling circuit for quantum proposals requires a k-qubit generalization of the R_{zz} gate, denoted as $R_{zzz\ldots z}$. Here, k represents the order of interactions allowed in the surrogate network, with $R_{zzz\ldots z}$ acting on k-qubits. This section outlines two methods to implement these gates using different kinds of two-qubit entangling gates along with arbitrary single-qubit unitary gates.

CNOT gate + arbitrary single qubit unitary operations : The first approach utilizes a gate set composed of the CNOT gate and arbitrary single-qubit rotations. The decomposition of the $R_{zzz\ldots z}$ gate into this gate set is as follows:

$$
R_{zzz..z}(\theta) = \prod_{i=0}^{k-2} CNOT(i, i+1) \;\; R_z^k(\theta) \;\; \prod_{i=k-2}^{0} CNOT(i, i+1)
$$

where the CNOT gates create the necessary entanglement structure, and the $R_z^k(\theta)$ gate applies the z-rotation on the k-th qubit. This decomposition is depicted in Figure S4.

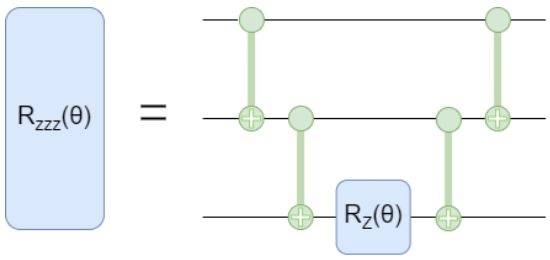

FIG. S4. Decomposition of $R_{zzz\ldots z}$ gate into CNOT gates and arbitrary single-qubit rotations.

Echoed Cross-Resonance (ECR) gate + arbitrary single qubit unitary operations: An alternative implementation of the $R_{zzz\ldots z}$ gate utilizes the Echoed Cross-Resonance (ECR) gate as the two-qubit entangling operation, combined with arbitrary single-qubit rotations. The ECR gate is functionally equivalent to the CNOT gate, differing only by additional single-qubit rotations. However, they are typically preferred over CNOT gates as they are native interactions in certain quantum architectures and often exhibit lower error rates.

A known relationship between the CNOT and ECR gates allows us to express the $R_{zzz\ldots z}$ gate in terms of ECR gates[23]. Specifically, the relation is:

$$\text{CNOT} = \left[R_Z(-\pi/2) \otimes R_Z(-\pi)\sqrt{X}\,R_Z(-\pi)\right] \text{ECR}\,[X \otimes I]\,,$$

where $R_Z(\theta)$, $\sqrt{X}$, and X are single-qubit operations. This relation neglects a global phase of $\pi/2$. Using this equivalence, the $R_{zzz\ldots z}$ gate can be decomposed into a sequence of ECR gates and arbitrary single-qubit rotations. The decomposition is illustrated in Figure S5.

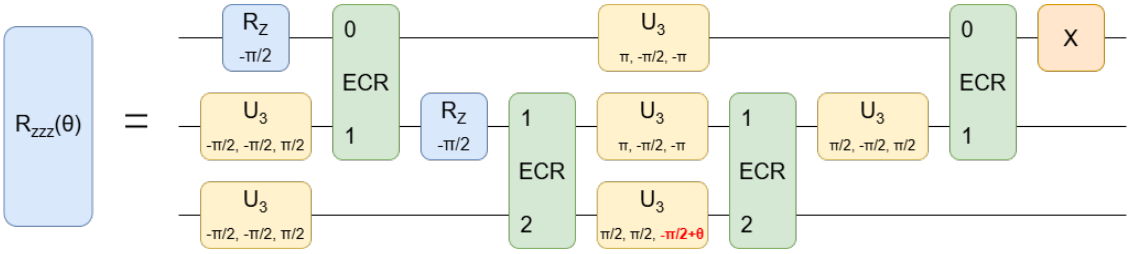

FIG. S5. Decomposition of $R_{zzz\ldots z}$ gate into ECR gates and arbitrary single-qubit rotations; correct up to a global phase. Here, $U3$ gate is the universal single-qubit rotation gate parameterized by three angles θ, ϕ, and λ, and its matrix form is given by Eq.S20 below

$$U3(\theta,\phi,\lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{pmatrix} \tag{S20}$$

Implementing the k-qubit $R_{zzz\ldots z}$ gates on quantum hardware requires only $2k$ two-qubit entangling gates. This linear scaling in the number of entangling gates ensures that the circuits remain efficient and practical for hardware implementation, even as k increases. Additionally, the use of hardware-native gate sets, such as ECR gates, further enhances the performance by minimizing gate overhead and reducing error rates in architectures where these gates are natively supported.

S7. ENERGY ERRORS VS ITERATIONS

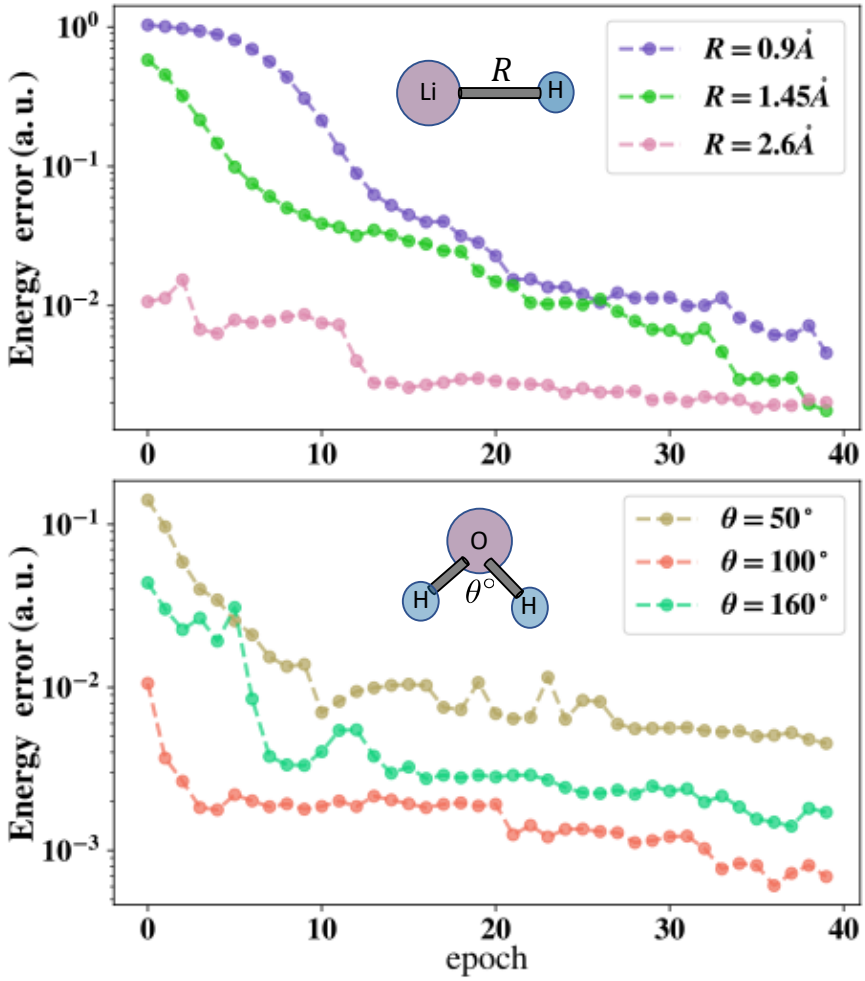

FIG. S6. The energy errors vs iterations (epoch) obtained during training graph G_1 (defined in main text) to approximate the ground state of H_2O, LiH as a function of geometrical parameters in the system (defined in inset). We use not more than 40 iterations for each system let zero-variance extrapolation decrease the energies further.

S8. RESOURCE REQUIREMENTS FOR BENCHMARKS

This section details the parameter counts for the RBM-NQS or its 2-surrogate, circuit evaluations (query complexity of the circuit), parameter counts of the circuit for one training instance for all the benchmark systems we have studied in this work. To provide easy comparisons, we shall also display what those quantities would be in the more general case of an n-qubit quantum system where the physical dimension of the accessible Hilbert space is 2^n. As a result at the beginning of the table all quantities are expressed in terms of the n to illustrate size-dependant scaling.

TABLE S1. Parameter counts and sampling cost across physical systems at representative operating points.

System	Point	Size	Dim	RBM Params	2-Surrogate Params	Circuit Params	Queries
Asymptotic Scaling	–	n	2^n	$2(nm+n+m)$	$n+\frac{n(n-1)}{2}$	$2n+\frac{n(n-1)}{2}$	$\sim n^{\ll 1}$
XXZ $(n=8, m=8)$	$\Delta/J=-1$	8	256	160	36	44	10,000
LiH $(n=8, m=8)$	$R_{\text{Li-H}}=1.45$ Å	8	256	160	36	44	10,000
	$R_{\text{Li-H}}=2.6$ Å	8	256	160	36	44	10,000
H_2O $(n=12, m=12)$	$\angle$H−O−H $=100°$	12	4096	336	78	90	10,000
	$\angle$H−O−H $=160°$	12	4096	336	78	90	10,000

Legend.

- **Size**: number of qubits n.
- **Dim**: dimension of the Hilbert space, $\dim(V)=2^n$.
- **RBM Params**: total number of parameters in the original NQS or Restricted Boltzmann machine $(\vec{a}, \vec{b}, \vec{W})$.
- **2-Surrogate Params**: parameters of the constructed surrogate NQS $(\vec{l}, \vec{J})$.
- **Circuit Params**: number of tunable angles in the quantum circuit constructed from the surrogate.
- **Queries**: This is the total number of circuit calls or samples $N_{\text{samples}} \approx 10,000$ extracted and is independent of n at each training instance (More on this in next point)

One must note in the top row of the Table S1 above, m is an auxiliary control knob determining the number of hidden neurons in the RBM ansatz, **typically** $m=n$. Since the quantum circuit is built from the surrogate model, the circuit parameters correspond to n single-qubit $R_z(\vec{\theta}_z)$ gates with $\vec{\theta}_z \propto \vec{l}$ and $\binom{n}{2}$ two-qubit $R_{zz}(\vec{\theta}_{zz})$ gates with $\vec{\theta}_{zz} \propto \vec{J}$. A single layer of n single-qubit R_x gates is parameter independent. This count is for one Trotter layer, and the same parameter set is reused for all N_{trot} layers, so the total number of independent parameters remains $2n+\frac{n(n-1)}{2}$. For all benchmarks $N_{\text{trot}} \sim 50$ but no new parameters are added from one layer to another. Also, these parameters are not variationally optimized but are fixed by the induced surrogate instance $(\vec{l}, \vec{J})$. In our simulations $N_{\text{samples}} \approx 10{,}000$, analogous to the total number of measurement shots used for estimating expectation values, and does not grow strongly with system size n. Each sample returns a bit-string $\vec{v}$. Across all benchmarks, each training instance required approximately 30-40 optimization iterations (ZNE decreased energies further), with total wall-clock training times of order a few minutes on a standard desktop CPU (Apple M1, 16 GB RAM), confirming that surrogate fitting and circuit sampling remain computationally inexpensive. The total physical training time of the algorithm is of course contingent on the hardware specifications (RAM, processor speed) being used i.e. the kind of CPU/GPU etc.

S9. RUNTIME AND ACCURACY BENCHMARKS ON REAL QUANTUM HARDWARE FOR LARGER SYSTEM SIZES

In this section we shall focus on additional computations to identify regimes where the protocol is advantageous and to clarify its current limitations at system sizes far beyond the original benchmarks displayed in main text. To probe such regimes in system sizes, we study 10 randomly generated sparse 2-surrogate instances $(\vec{l}, \vec{J})$ with nearest-neighbor connectivity, which are compatible with current hardware constraints for upto $n \leq 32$ qubits. We compare the runtime required to generate individual bit-strings using classical CPU/GPU based implementations of our quantum-circuit based sampler and also a direct QPU-based implementation. We use $ibm - fez$ and $ibm - pittsburgh$ for all computations. The results are displayed in Fig.S7. As is clear from the plots, the raw wall time cost of classically running the circuit in CPU/GPU grows exponentially with system size n for late values of $n \geq 20$. In contrast, the QPU runtime remains horizontally flat on the log-scale axis, implying $\log T_{\mathrm{QPU}}(n) \approx$ constant and therefore $T_{\mathrm{QPU}}(n) = O(1)$. It remains approximately constant at ~ 5 s per sample up to $n \leq 32$. In addition to runtime, in Fig.S8 , we quantify sampling quality by measuring the ℓ_2-norm difference between the empirically reconstructed surrogate distribution ϕ_{sample} and the exact target distribution ϕ_{target} for 50 instances of $(\vec{l}, \vec{J})$. For the quantum-enhanced sampler, the error remains below 0.1 using only ~ 4000 samples across all tested system sizes, whereas classical uniform sampling requires substantially more samples (even 10,000 is not enough) and exhibits an exponentially growing error. These results complement the earlier mixing-time analyses reported for $n \leq 10$ and demonstrate a practical and direct display of sampling advantage beyond the asymptotic claims.

The primary limitations currently arise from hardware connectivity constraints and communication latency between classical controllers and QPU backends. For general RBM-induced surrogate models, the coupling matrix $\vec{J}$ is typically dense, leading to nonlocal two-qubit gates that require routing and SWAP operations on hardware with restricted connectivity, thereby increasing circuit depth and noise and even runtime. For benchmark examples studied in this work, the surrogate model was non-sparse which precluded direct hardware implementation with the accessible fleet of devices. One must note other quantum computing platforms like newly launched Helios from Quantinuum [24] does have 98 qubits with all-to-all connectivity and superior noise profiles of single and two-qubit gates (99.9975 % for single qubit ones and 99.921 % for two-qubit gates) which can be leveraged for such computations in the future. This offers a natural path forward for deployment. For latency bottlenecks in the workflow, instead of naive sequential execution of one circuit per MCMC step, one can amortize communication through batched sampling, wherein a single circuit execution returns many bit-strings (shots) that advance multiple MCMC steps as parallel chains. The proposal circuit can be compiled once and reused with updated parameters, and refreshed only periodically rather than at every step. When available, hybrid runtime or near-QPU classical execution further removes network round trips by collocating the MCMC loop with the QPU backend. Additionally, algorithmic variants such as multiple-try Metropolis [25] or delayed-acceptance schemes can leverage proposal pools from each circuit call [26] while preserving detailed balance, further reducing feedback frequency. Thus for denser models, straightforward extensions using these mitigation strategies enable near-term demonstrations on larger systems as hardware connectivity and runtime infrastructure continue to improve. Taken together, these findings indicate that the advantages of the proposed quantum-enhanced sampling protocol are not merely asymptotic but can be practically realized beyond the presently implemented benchmark system sizes.

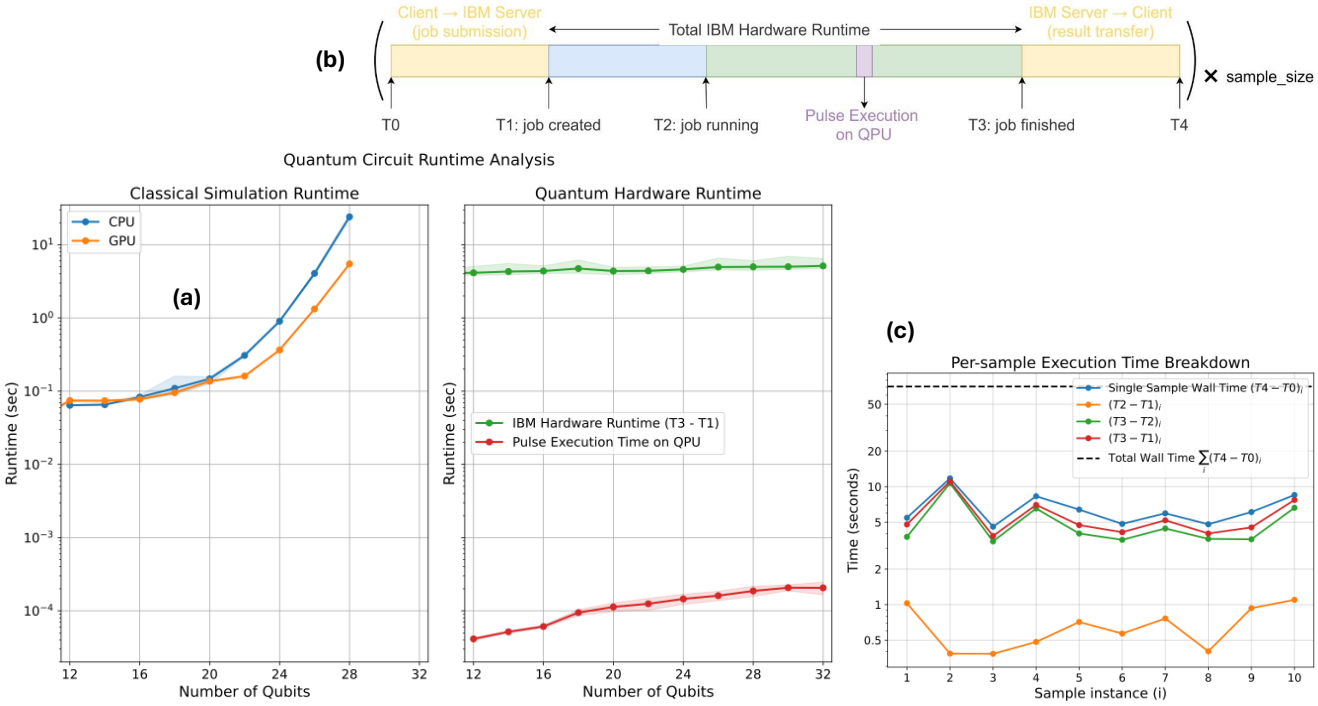

FIG. S7. (a) The runtime cost(raw wall time) for generating a single bit-string sample using the quantum-enhanced sampler (Proposal H) using a CPU (blue), GPU(orange) and QPU(green) vs system size upto $n \leq 32$. Also added is the raw pulse execution time window on QPU (see (b)) below. For all computations $ibm - fez$ and $ibm - pittsburgh$ is used. It is clear that the runtime cost is insensitive to system size for QPU backend but blows exponentially when the circuit instance is executed over CPU/GPU especially beyond $n \geq 20$. The shaded regions in each curve corresponds to interquartile range (first and 3rd quartile) for data obtained when the circuit is executed for 10 instances of 2-surrogate parameter $(\vec{l}, \vec{J})$ and for each such parameter instance, 10 bit-string is sampled. The raw points on the curve is the median (2nd quartile) of such instances (b) A schematic for the breakdown of the time window obtained from metadata during IBMQ execution. The green band corresponds to exact circuit execution window when IBMQ cloud platform locks the device for the user. (c) A detailed breakdown of raw runtime analysis (for various time windows in (b)) for $n = 6$ qubits when 10 bit-strings are sampled from the quantum circuit . Sample to sample fluctuations due to communication latencies are captured.

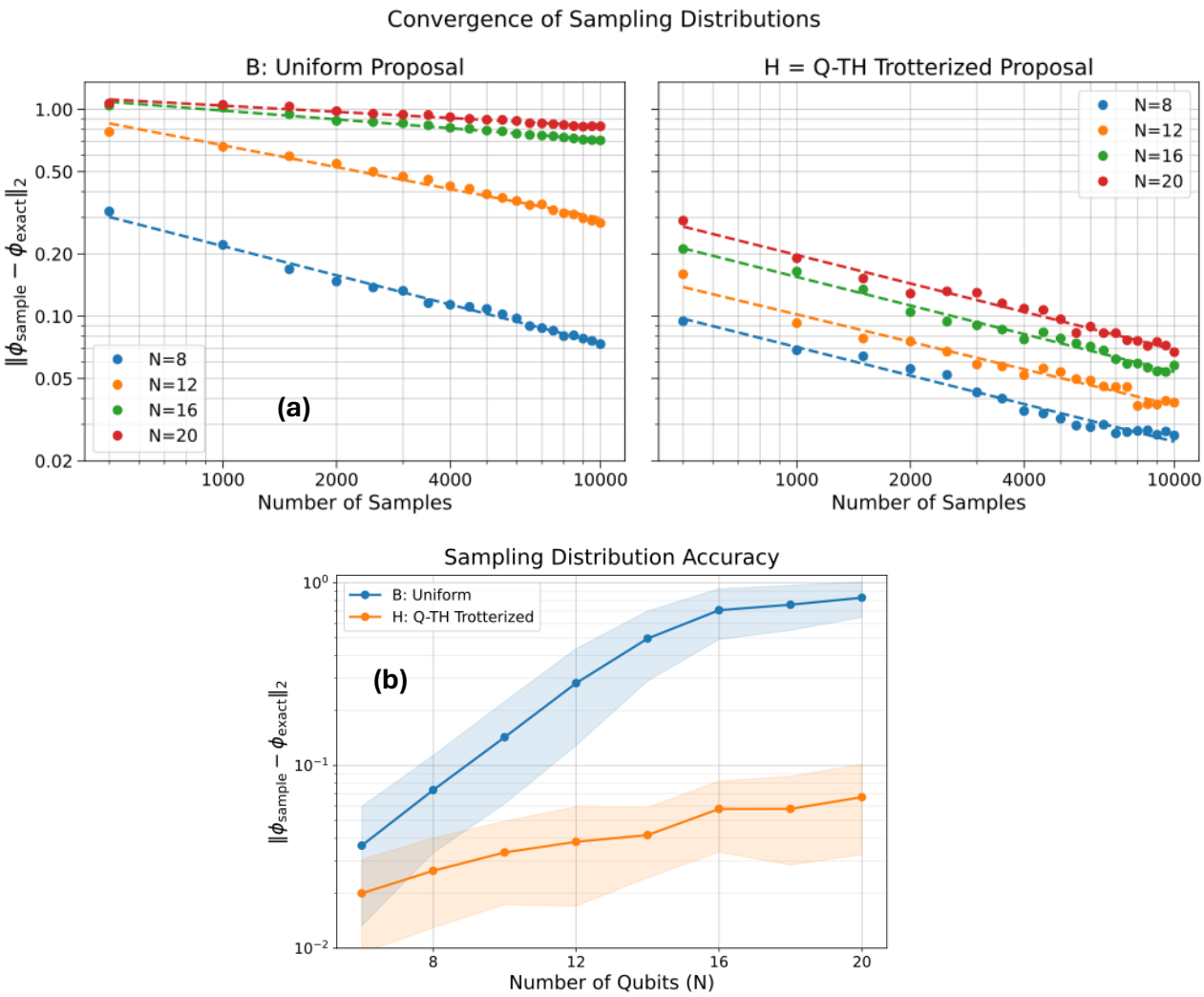

FIG. S8. (a) ℓ_2-norm difference between the empirically constructed 2-surrogate distribution ϕ_{sample} and the exact target distribution ϕ_{exact} averaged over 50 parameter instances $(\vec{l}, \vec{J})$ as a function of number of samples extracted for various system sizes. This is done for two sampling strategies. (left) A commmonly used classical sampling proposal (Proposal B: Uniform) is used and (right) the quantum-enhanced MCMC protocol (Proposal H : Q-TH ($\tau = 11, \gamma = 0.425$)-Trotter) is used. In each case convergence is displayed as a function of number of samples drawn for various system sizes $n \leq 20$. We see clearly that for Proposal H which is the quantum-enhanced one, convergence happens must faster and even at 4000 samples the error in the full distribution is below a threshold of 0.1 for all system sizes. The uniform classical sampling strategy cannot reach the threshold even with 10,000 samples for any system size beyond $n \geq 8$ (b) The final converged $||\phi_{sample} - \phi_{target}||_2$ obtained from (a) after 10,000 iterations is plotted for different system sizes for both the sampling strategies (Proposal B and quantum -enhanced Proposal H). It is clear that Proposal H, the error grows as sub-exponentially while for the classical proposal B it is indeed exponential. The variance showing parameter-induced fluctuations is also shaded in each proposal. The plateauing effect for the uniform sampler (blue) is due to the fact that ℓ_2-norm differences of stochastic vector is upper bounded by $\sqrt{2}$. This along with Fig.3 in main manuscript and Fig.S2 and Fig.S3 in Section S4 and the figure for runtime analysis (Fig.S7) above clearly demonstrates that quantum-sampling is not only advantageous in terms of superior quality of samples but is also efficient in raw execution walltime even at enlarged sizes which makes the protocol doubly lucrative. The shaded region shows interquartile ranges for 50 instances of $(\vec{l}, \vec{J})$ per system size.

S10. WHY THE QUANTUM PROPOSAL WORKS BETTER?

In this section we aim to understand the features that make the quantum proposal (H) perform better than the classical counterparts i.e. local proposal (A), Uniform proposal (B), Haar random proposal (C) in main text. The primary difference can be attributed to the fact that the quantum-enhanced priors contain information about the target distribution $\phi(\vec{l}(\vec{X}), \vec{J}(\vec{X}), \vec{v})$. This

is due to the fact that the surrogate G_2 we define for graph G_1 in the main text encodes a distribution $\phi(\vec{l}(\vec{X}), \vec{J}(\vec{X}), \vec{v})$ defined as

$$\phi(\vec{l}(\vec{X}), \vec{J}(\vec{X}), \vec{v}) = exp(-\beta(\sum_i l_i(\vec{X})v_i + \sum_{ij} J_{ij}(\vec{X})v_i v_j))$$

. Now we can look at the description of the circuit in previous section and in Section III: Algorithm of the main text wherein it is clearly specified that the quantum circuit of proposal H is essentially sampling from the distribution $P_{prop}(\vec{v}^{(i+1)}|\vec{v}^{(i)}) = |\langle\vec{v}^{(i+1)}|U(\tau = 2, \gamma = 0.425)|\vec{v}^{(i)}\rangle|^2$ with $U(\tau,\gamma) = (e^{-i(1-\gamma)h_2\delta t}e^{-i\gamma h_1\delta t})^{N_{trot}}$ and $N_{trot} = \frac{\tau}{\delta t}$. It must be noted that $h_1 = \sum_i l_i(\vec{X})\sigma_z^{(i)} + \sum_{i,j} J_{ij}(\vec{X})\sigma_z^{(i)}\sigma_z^{(j)}$ and h_2 is a mixer defined as $h_2 = \sum_i \sigma_x^{(i)}$. The hyperparameters (γ, τ) in other quantum-enabled proposals like in D are randomly sampled as or are kept at fixed values (see main text). Thus the information about the sampling distribution $\phi(\vec{l}(\vec{X}), \vec{J}(\vec{X}), \vec{v})$ is baked into the proposal distribution through h_1 above. The very nature of the functional form for $\phi(\vec{l}(\vec{X}), \vec{J}(\vec{X}), \vec{v})$ dictates that significant number of samples will be drawn for configurations which corresponds to low values of h_1. We shall see that the quantum-enabled proposals sample configurations which are even though of lower energy content with respect to h_1 but may have very different Hamming distances. This is not the case for local proposal (A) wherein a single-site mutation can only generate configurations with nearest Hamming distances exclusively. As a result, if there are other configurations of similar energy but of very different Hamming distances, it might take a lot of samples to generate them. Besides for uniform (B) and Haar random proposal (C), they lack any information about the geography of the configuration landscape and is problem-agnostic completely. As a result, subsequent configurations chosen by that proposal is not guided by energy differences in the configuration landscape and hence is amenable to rejection by the acceptance criteria. Once trapped, this may lead to possible over-population of configurations corresponding to any local minima. All these points are clarified by the following data which has been computed for a system of n=m=8 (8 visible neurons and 8 hidden neurons). The Hamming distances and energies of each configuration corresponding to $\phi(\vec{l}(\vec{X}), \vec{J}(\vec{X}), \vec{v})$ has been computed by using randomly generated $(\vec{l}, \vec{J})$ with each component of either vector being sampled from a standard normal distribution.

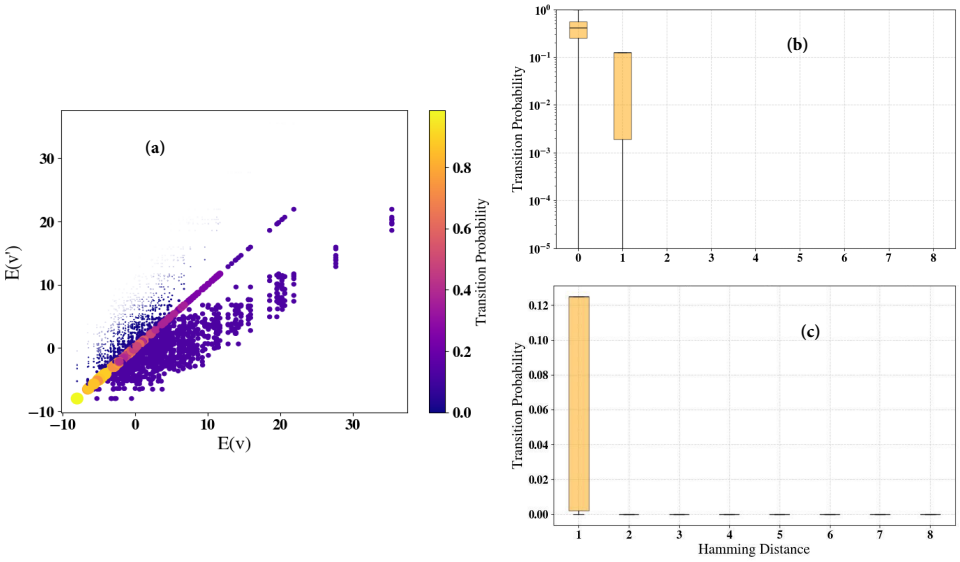

FIG. S9. **Energy and Hamming distance profile of local proposal (A)** (a) Energy values $E(\vec{v}) = h_1$ for configurations $\vec{v}$ are arranged in a matrix for comparison with energies of candidate configurations $E(\vec{v}')$ for the local proposal (A) in main text. Points $(E(\vec{v}), E(\vec{v}'))$ are colored based on transition probabilities $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ (Eq. 9). The proposal strongly favors $\vec{v}'$ with similar or lower energy due to the acceptance criteria in main text. This will be seen to be common to all proposals. (b) The Hamming distance between $\vec{v}$ and $\vec{v}'$ is plotted on the x-axis against transition probabilities $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Given multiple configurations can be at a given Hamming distance from a chosen configuration, the median, interquartile range, and extreme transition probabilities of all such configurations are shown. The local proposal only generates configurations differing by at most one Hamming unit. (c) As in (b), but plotted on a linear scale starting at a Hamming distance of 1, excluding dominant transitions at zero Hamming distance for clarity.

For the Haar random proposal, from the plots in Fig.S11 and in Fig.3(b) in main text it is clear that Haar random unitary performs like the uniform proposal (see Fig.S10, Fig.S11 and Fig.3(b) in main text). This is despite the fact that the quantum state generated from the Haar-random unitary is highly entangled and exact construction of the unitary requires exponential gate resources in system size. We provide a mathematical justification for this observation below through a rigorous proof

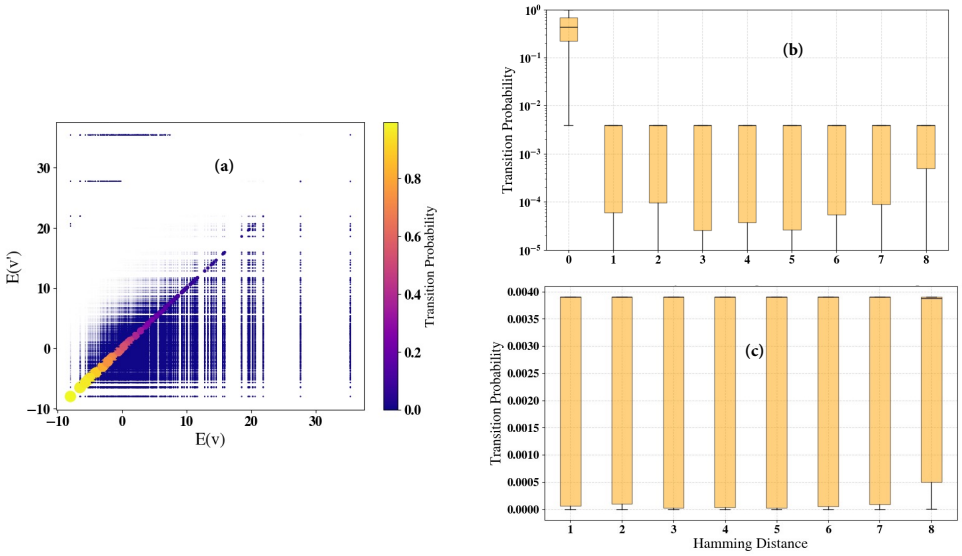

FIG. S10. **Energy and Hamming distance profile of uniform proposal (B)**(a) Energy values $E(\vec{v}) = h_1$ for configurations $\vec{v}$ are arranged in a matrix for comparison with energies of candidate configurations $E(\vec{v}')$ for the uniform proposal (B) in main text. Points $(E(\vec{v}), E(\vec{v}'))$ are colored based on transition probabilities $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ (Eq. 9). The proposal favors $\vec{v}'$ with similar or lower energy more as seen before (b) The Hamming distance between $\vec{v}$ and $\vec{v}'$ is plotted on the x-axis against transition probabilities $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Given multiple configurations can be at a given Hamming distance from a chosen configuration, the median, interquartile range, and extreme transition probabilities of all such configurations are shown. The uniform proposal can generate configurations differing by many Hamming distance units. (c) As in (b), but plotted on a linear scale starting at a Hamming distance of 1, excluding dominant transitions at zero Hamming distance for clarity. It is clear that the maximum transition probability of all transitions differing by hamming units are at most 0.004.

Lemma 6. *Given a distribution of unitaries* $U \sim P_{Haar}(U)$ *where* $P_{Haar}(U)$ *is the Haar measure, then* $\mathbb{E}_{U\sim P_{Haar}(U)}[|\langle\vec{v}'|U|\vec{v}\rangle|^2] = \frac{1}{2^n}\ \forall\ \vec{v},\vec{v}' \in \{0,1\}^n$

Proof.

$$\mathbb{E}_{U\sim P_{Haar}(U)}[\sum_{\vec{v}'\in\{0,1\}^n} |\langle\vec{v}'|U|\vec{v}\rangle|^2] = 1 \qquad \vec{v}\in\{0,1\}^n$$

$$\Rightarrow \sum_{\vec{v}'\in\{0,1\}^n} \mathbb{E}_{U\sim P_{Haar}(U)}[|\langle\vec{v}'|U|\vec{v}\rangle|^2] = 1$$

$$\Rightarrow \sum_{\vec{v}'\in\{0,1\}^n} \mathbb{E}_{U\sim P_{Haar}(U)}[|\langle\vec{0}|V_{\vec{v}'}^{\dagger}U|\vec{v}\rangle|^2] = 1 \qquad V_{\vec{v}'}|\vec{0}\rangle = |\vec{v}'\rangle$$

$$\Rightarrow \sum_{\vec{v}'\in\{0,1\}^n} \mathbb{E}_{U\sim P_{Haar}(U)}[|\langle\vec{0}|V_{\vec{v}'}^{\dagger}UW_{\vec{v}}|\vec{0}\rangle|^2] = 1 \qquad W_{\vec{v}}|\vec{0}\rangle = |\vec{v}\rangle$$

$$\Rightarrow \sum_{\vec{v}'\in\{0,1\}^n} \mathbb{E}_{U\sim P_{Haar}(U)}[|\langle\vec{0}|U'|\vec{0}\rangle|^2] = 1 \qquad U' = V_{\vec{v}'}^{\dagger}UW_{\vec{v}}$$

$$\Rightarrow \mathbb{E}_{U'\sim P_{Haar}(U')}[|\langle\vec{0}|U'|\vec{0}\rangle|^2] \sum_{\vec{v}'\in\{0,1\}^n} 1 = 1 \qquad U'\sim U\sim P_{Haar}(U) = P_{Haar}(U')\ \because \text{unitary invariance}[27--29]$$

$$\Rightarrow \mathbb{E}_{U'\sim P_{Haar}(U')}[|\langle\vec{0}|U'|\vec{0}\rangle|^2] = \frac{1}{2^n}$$

Now $\mathbb{E}_{U'\sim P_{Haar}(U')}[|\langle\vec{0}|U'|\vec{0}\rangle|^2] = \mathbb{E}_{U\sim P_{Haar}(U)}[|\langle\vec{0}|V_{\vec{v}'}^{\dagger}UW_{\vec{v}}|\vec{0}\rangle|^2] = \mathbb{E}_{U\sim P_{Haar}(U)}[|\langle\vec{v}'|U|\vec{v}\rangle|^2]$ due to left and right unitary invariance [27–29]. Thus, it is clear from the above proof that $\mathbb{E}_{U\sim P_{Haar}(U)}[|\langle\vec{v}'|U|\vec{v}\rangle|^2] = \frac{1}{2^n}$ □

Lemma 7. *Given a distribution of unitaries* $U \sim P_{Haar}(U)$ *where* $P_{Haar}(U)$ *is the Haar measure, then the following is true:*

i $\lim_{n\to\infty} Var(|\langle\vec{v}'|U|\vec{v}\rangle|^2) = O(2^{-2n})\ \forall\ \vec{v},\vec{v}' \in \{0,1\}^n$

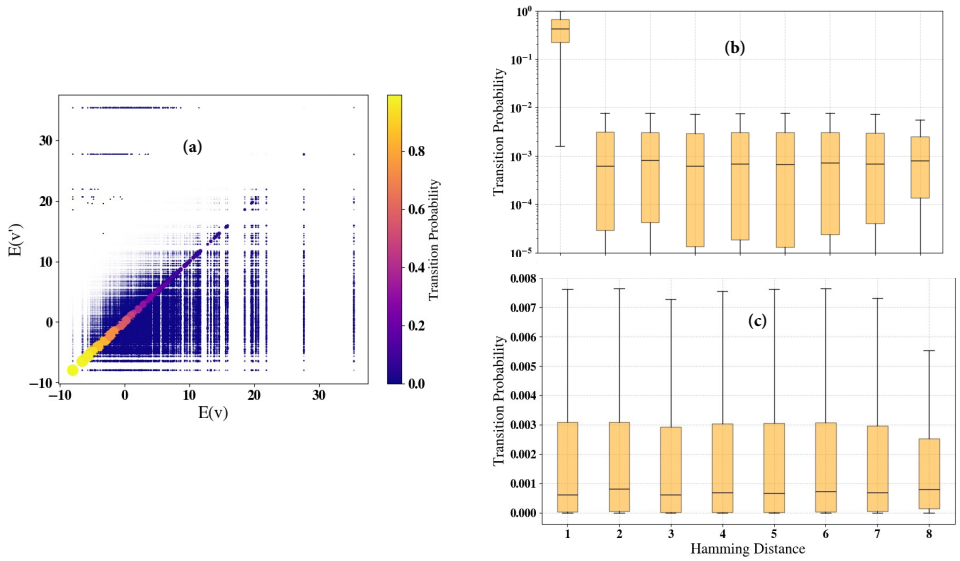

FIG. S11. **Energy and Hamming distance profile of Haar-random proposal (C)**(a) Energy values $E(\vec{v}) = h_1$ for configurations $\vec{v}$ are arranged in a matrix for comparison with energies of candidate configurations $E(\vec{v}')$ for the Haar random proposal (C) in main text. Points $(E(\vec{v}), E(\vec{v}'))$ are colored based on transition probabilities $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ (Eq. 9). The proposal favors $\vec{v}'$ with similar or lower energy more as seen before for the uniform proposal (b) The Hamming distance between $\vec{v}$ and $\vec{v}'$ is plotted on the x-axis against transition probabilities $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Given multiple configurations can be at a given Hamming distance from a chosen configuration, the median, interquartile range, and extreme transition probabilities of all such configurations are shown. The Haar random proposal can generate configurations differing by many Hamming distance units. (c) As in (b), but plotted on a linear scale starting at a Hamming distance of 1, excluding dominant transitions at zero Hamming distance for clarity. It is clear that the Haar random proposal does better than the uniform proposal as the maximum transition probability of all transitions differing by many hamming units can be at most 0.008 with the median close to 0.001.

ii Any deviation of a typical instance $|\langle\vec{v}'|U|\vec{v}\rangle|^2$ *from the mean* $\mathbb{E}[|\langle\vec{v}'|U|\vec{v}\rangle|^2] = \frac{1}{2^n}$ *(see Lemma 6) is exponentially suppressed in the size of the system n i.e.* $P(|\langle\vec{v}'|U|\vec{v}\rangle|^2 - \frac{1}{2^n}| \geq \epsilon) \leq 2\, e^{-2^n c\epsilon^2}$ *where* $c, \epsilon \in \mathbb{R}_+$

Proof. (i) To prove the first part of the above statement, we use the basic definition of a $n = log_2(d)$ qubit unitary matrix U sampled from the Haar ensemble which is that every column of the unitary comprises of entries which are uniform on the unit sphere in $\mathbb{C}^d$. One of the easiest ways to generate the matrix is to sample the elements as i.i.d from a complex gaussian ($\mathcal{N}_c(0, \mathbb{I})$)[29, 30], followed by QR factorization to ensure unitarity. Each such element in a given column (say the kth column) can be thus modeled as $U[:, k] = \frac{\vec{g}_k}{||\vec{g}_k||_2}$ where $\vec{g}_k \sim \mathcal{N}_c(0, \mathbb{I}) \quad \forall k$. Now if each $\vec{g}_k \sim \mathcal{N}_c(0, \mathbb{I})$ then that essentially means that $Re(g_k^{(i)}) \sim \mathcal{N}(0, \frac{1}{2})$ and independently $Im(g_k^{(i)}) \sim \mathcal{N}(0, \frac{1}{2})$ for every i-th element (i being the row index) in the column k. The joint-distribution $p(Re(g_k^{(i)}), Im(g_k^{(i)})) \sim e^{-Re(g_k^{(i)})^2 - Im(g_k^{(i)})^2}$. It is easy to show by change of variables to $(|g_k^{(i)}|^2 = Re(g_k^{(i)})^2 + Im(g_k^{(i)})^2, Arg(g_k^{(i)}) = tan^{-1}\frac{Im(g_k^{(i)})}{Re(g_k^{(i)})})$ through a Jacobian transformation and integrating out $Arg(g_k^{(i)})$ that $p(|g_k^{(i)}|) \sim |g_k^{(i)}|e^{-|g_k^{(i)}|^2}$. Finally through another Jacobian transformation of co-ordinates one can establish that $P(y = |g_k^{(i)}|^2) \sim \frac{1}{2\sqrt{y}}e^{-y}\mathbb{I}_{y\geq 0}$ where $\mathbb{I}_{y\geq 0}$ is an indicator variable ensuring that y is defined for only non-negative values as physically it corresponds to $y = |g_k^{(i)}|^2$. Now, we are interested

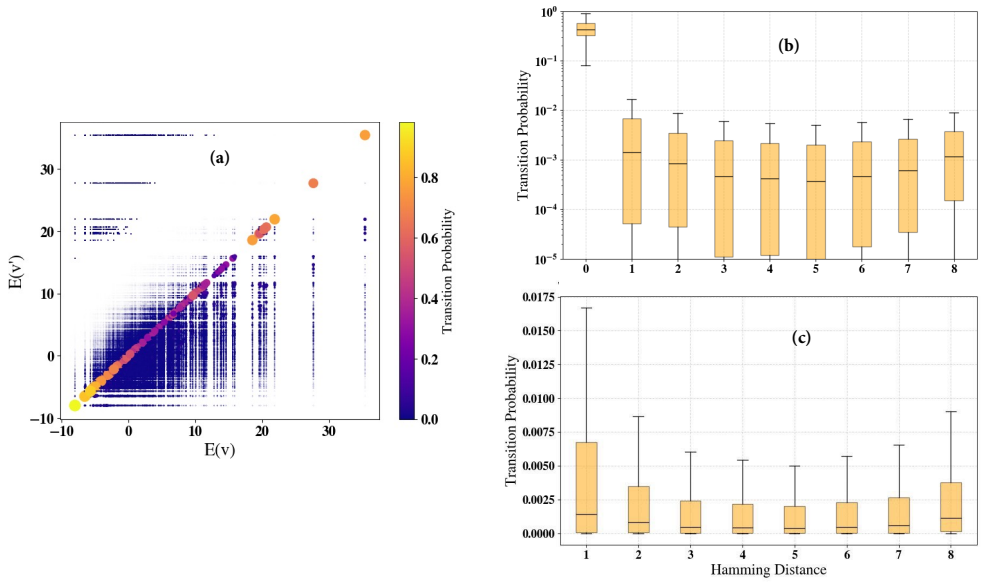

FIG. S12. **Energy and Hamming distance profile of quantum-enabled Trotterized proposal (H)**(a) Energy values $E(\vec{v}) = h_1$ for configurations $\vec{v}$ are arranged in a matrix for comparison with energies of candidate configurations $E(\vec{v}')$ for Quantum time-homogeneous Trotterized Proposal (H) in main text. Points $(E(\vec{v}), E(\vec{v}'))$ are colored based on transition probabilities $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$ (Eq. 9). The proposal favors $\vec{v}'$ with similar or lower energy more as seen before for the uniform or Haar random proposal before (b) The Hamming distance between $\vec{v}$ and $\vec{v}'$ is plotted on the x-axis against transition probabilities $T(\vec{v}^{(i+1)}|\vec{v}^{(i)})$. Given multiple configurations can be at a given Hamming distance from a chosen configuration, the median, interquartile range, and extreme transition probabilities of all such configurations are shown. The quantum-enabled proposal (H) can also generate configurations differing by many Hamming distance units. (c) As in (b), but plotted on a linear scale starting at a Hamming distance of 1, excluding dominant transitions at zero Hamming distance for clarity. It is clear that the quantum-enabled proposal does even better than the uniform or Haar random proposal as the maximum transition probability of all transitions differing by many hamming units can be as high as 0.0175 with the median close to 0.002. Besides even configurations differing by a Hamming unit of 8 (highest than be achieved for n=8) also has an accessible transition probability with a maximum close to 0.01 and the median close to 0.002. This is not seen in any of the proposals above i.e. the local (A), uniform (B) or Haar random (C). The Haar random one is the closest reporting a maximum transition probability of 0.0055 and a median close to 0.001 for such transitions. The uniform does poorer than that and the local proposal cant even make such jumps. This feature of quantum proposal was also noted in Ref[4].

in finding joint distribution of $\Phi(|U[:,k]|^2)$ for all elements in column k which can be found as follows

$$
\begin{aligned}
\Phi(|U[:,k]|^2) = \; & = \Pi_{i=1}^{d} \phi(|U[i,k]|^2) \\
& = \Pi_{i=1}^{d} \phi\left(\frac{|g_k^{(i)}|^2}{\sum_j |g_k^j|^2}\right) \\
& \equiv \Pi_{i=1}^{d-1} P(\underbrace{\sum_j |g_k^{(j)}|^2 |U[i,k]|^2}_{|g_k^{(i)}|^2}) \times P(\underbrace{\sum_j |g_k^{(j)}|^2 (1 - \sum_{m=1}^{d-1} |U[m,k]|^2)}_{|g_d^{(i)}|^2}) \quad [29,\ 31] \\
& = \Pi_{i=1}^{d-1} P(y_i = S|U[i,k]|^2) \times \underbrace{P(y_d = S(1 - \sum_{m=1}^{d-1} |U[m,k]|^2))}_{|g_n^{(i)}|^2} \qquad S = \sum_j |g_k^{(j)}|^2 \in \mathbb{R}_+ \\
& \propto \Pi_{i=1}^{d-1} e^{-S|U[i,k]|^2} \times e^{-S\sum_{m=1}^{d-1}(1-|U[i,k]|^2)} \times S^{d-1} \times \mathbb{I}_{S|U[i,k]|^2 \geq 0, \sum_m |U[m,k]|^2 = 1} \qquad \because\ det(J) = S^{d-1} \\
& \propto e^{-S} \times S^{d-1} \times \mathbb{I}_{S|U[i,k]|^2 \geq 0, \sum_m |U[m,k]|^2 = 1} \\
& \propto e^{-S} \times S^{d-1} \times \mathbb{I}_{S \geq 0} \times \mathbb{I}_{\sum_i^{d-1} |U[i,k]|^2 \leq 1}
\end{aligned}
\tag{S21}
$$

where in the last line we have used indicator variables for $\{|U[i,k]|^2\}_{i=1}^{d-1}$ as those are the only independent elements. For a given choice of $\{|U[i,k]|^2\}_{i=1}^{d-1}$ the element $|U[d,k]|^2$ is fixed as, $U[d,k] = 1 - \sum_{m<=d-1} |U[m,k]|^2$. This also makes $\Phi(|U[1,k]|^2....|U[d-1,k]|^2, S)$ i.e. Φ is functionally dependant only on $\{|U[i,k]|^2\}_{i=1}^{d-1}$ and $S = \sum_j |g_k^{(j)}|^2$. Now that we have the full joint distribution of $\Phi(|U[1,k]|^2....|U[d-1,k]|^2, S)$, it is possible to get a marginal distribution $\phi(|U[1,k]|^2)$ by integrating over all other variables. This can be done as follows:

$$\begin{aligned}
\Phi(|U[1,k]|^2....|U[d-1,k]|^2, S) &\propto e^{-S} \times S^{d-1} \times \mathbb{I}_{S\geq 0} \mathbb{I}_{\sum_i^{d-1} |U[i,k]|^2 \leq 1} \\
\Phi(|U[1,k]|^2....|U[d-1,k]|^2, S) &\propto \underbrace{\frac{e^{-S} \times S^{d-1}}{\Gamma(d)}}_{\Gamma-\text{dist}(d,1)} \times \underbrace{\Gamma(d) \mathbb{I}_{S\geq 0} \mathbb{I}_{\sum_i^{d-1} |U[i,k]|^2 \leq 1}}_{\text{Dirichlet-dist}(1,1,1...1_n)} [32, 33] \\
\text{Thus, } \phi(|U[1,k]|^2) &\propto \int_C \Phi(|U[1,k]|^2....|U[d-1,k]|^2, S)\ dS\, d|U[2,k]|^2....d|U[d-1,k]|^2 \\
\phi(|U[1,k]|^2) &\propto \text{Beta}(1, d-1) [34, 35]
\end{aligned} \tag{S22}$$

where $C = \{(|U[m,k]|^2, S) | m = 1..d-1, \sum_m |U[m,k]|^2 \leq 1, S \geq 0\}$ and in the last line we have used the fact that marginalizing over components of a Dirichlet distribution gives a Beta distribution []. Given we have established that $\boldsymbol{\phi(|U[1,k]|^2) \sim \text{Beta}(1, d-1)}$, we can directly yield the Variance and Mean from standard results [34, 35] as follows

$$\begin{aligned}
\mathbb{E}[|U[1,k]|^2] &= \frac{1}{d} \\
Var(|U[1,k]|^2) &= \frac{d-1}{d^2(d+1)}
\end{aligned} \tag{S23}$$

Given $n = log_2(d)$, thus $d = 2^n$, substituting in the above equations we get

$$\begin{aligned}
\lim_{n\to\infty} Var(|U[1,k]|^2) = \lim_{d\to\infty} Var(|U[1,k]|^2) &= \lim_{d\to\infty} \frac{d}{d^2(d+1)} - \lim_{d\to\infty} \frac{1}{d^2(d+1)} \\
&= \lim_{d\to\infty} \frac{1}{d(d+1)} - \lim_{d\to\infty} \frac{1}{d^2(d+1)} \\
&\approx \lim_{d\to\infty} \frac{1}{d(d+1)} \\
&= \lim_{d\to\infty} O(d^{-2}) \\
&= O(2^{-2n}) \qquad d = 2^n
\end{aligned} \tag{S24}$$

Similarly $\mathbb{E}[|U[1,k]|^2] = \frac{1}{2^n}$ which recovers the result of Lemma 6. Now $Var(|\langle \vec{v}'|U|\vec{v}\rangle|^2) = Var(|\langle \vec{0}|W_{\vec{v}}^\dagger U V_{\vec{v}'}|\vec{0}\rangle|^2) = Var(|\langle \vec{0}|U|\vec{0}\rangle|^2) = Var(|U[1,1]|^2)$ due to unitary invariance of Haar measure (see Lemma 6). Thus substituting k=1 in Eq.S24 we get the required result. $Var(|\langle \vec{v}'|U|\vec{v}\rangle|^2) = O(2^{-2n})$.

(ii) To prove the second part, one can show from Levy's lemma for Lipschitz continuous functions [36, 37] (true in this case as $|U[1,k]|^2$ is Lipschitz with a Lipschitz constant of 2) and also use the fact that $|U[1,k]|^2 \in \mathbb{S}^{d-1} \subseteq \mathbb{R}^d$. We set the constants $c, \epsilon \in \mathbb{R}_+$ and then one can straightforwardly use the Lemma as

$$P(|U[1,k]|^2 - \mathbb{E}[|U[1,k]|^2]| \geq \epsilon) \leq 2\, e^{-2dc\epsilon^2/4} \tag{S25}$$

Substituting $d = 2^n$, the above equation indicates that a typical instance of $|U[1,k]|^2$ would be concentrated near the mean exponentially in system size. Since the mean $\mathbb{E}[|U[1,k]|^2] = \frac{1}{d} = \frac{1}{2^n}$ (see Lemma 6 also) and $|U[1,1]|^2 = |\langle \vec{v}'|U|\vec{v}\rangle|^2$ by unitary invariance of Haar measure, so combining all these we have

$$\begin{aligned}
P(|\langle \vec{v}'|U|\vec{v}\rangle|^2 - \mathbb{E}[|\langle \vec{v}'|U|\vec{v}\rangle|^2]| \geq \epsilon) &\leq 2\, e^{-2^n c\epsilon^2} \\
P(|\langle \vec{v}'|U|\vec{v}\rangle|^2 - \frac{1}{2^n}| \geq \epsilon) &\leq 2\, e^{-2^n c\epsilon^2}
\end{aligned} \tag{S26}$$

□

Conceptually Haar-typical pure states act like purifications of an infinite-temperature state. This is best easily seen from the fact that averaging the action of Haar random unitary generates the action of a depolarizing channel $\mathbb{E}_{U\sim Haar}[U\rho U^\dagger] = \frac{\mathbb{I}}{2^n}$. For a Haar-random state, every small subsystem is 'almost' maximally mixed: its Von Neumann entropy follows Page's bound on average with deviations exponentially small in subsystem size [38, 39]. Hence such states carry essentially no structured correlations to amplify 'desirable' bit-strings. In line with Aaronson [40] and Gael [41], genuine speedups require problem-tailored circuits where one has to customize the gate operations such that constructive interference is created between desirable outcomes and destructive interference between undesirable ones. On average, Haar-random circuits drive the state close to maximally mixed, washing out inter-subsystem correlations. Our quantum-enabled proposals (D–H) are target-aware global moves: they bias sampling toward the low-energy sector and can change many bits at once (large Hamming jumps; see Fig. S12). By contrast, Haar-random proposals draw uniformly over all configurations (see Fig. S11), ignoring energy and Hamming costs and geography of the landscape, which inflates MCMC rejection rates and wastes samples. Besides it is easy to see that a Haar-random unitary is farthest (upto a global phase) from the target unitary (quantum proposals D-H) on an average. Let us assume that we have access to $U \sim P_{Haar}(U)$ and V is the target unitary from our quantum proposal. The assertion can be seen using Hilbert-Schimdt inner product of the difference of the two unitary i.e. $||U-V||_F^2$ and averaging over many choices of U as follows:

$$\begin{aligned} E_{U\sim P_{Haar}(U)}||U-V||_F^2 &= E_{U\sim P_{Haar}(U)} Tr((U-V)^\dagger(U-V)) \\ &= E_{U\sim P_{Haar}(U)}[\frac{1}{2^n}] - E_{U\sim P_{Haar}(U)}[2Re(Tr(V^\dagger U)] \\ &= \frac{1}{2^n} \qquad \because E_{U\sim P_{Haar}(U)}[2Re(Tr(V^\dagger U)] = 0 \end{aligned} \tag{S27}$$

where we have used the fact that $V^\dagger U \sim P_{Haar}(U)$ due to left and right unitary invariance of Haar measure[27–29] and $E_{W\sim P_{Haar}(W)}[Tr(W)] = 0$.

Also, we must emphasize that high entanglement content of a quantum state does not always equal lack of classical simulability. For example cluster states generated from Clifford circuits can be highly entangled [42, 43]. Yet all of them are classically simulable due to the famous Gottesmann-Knill theorem [44, 45]. Besides it has been noted in several recent years that quantum speed-up in algorithms is not due to high entanglement but due to non-stabilizerness/magic [46, 47]. For example in Refs [47–49] even constant or shallow-depth circuits become hard to sample from classically once non-Clifford gates are present. Our ansatz uses arbitrary-angle Rzz and Rz rotations (not restricted to multiples of $\frac{\pi}{2}$), hence it is non-Clifford by construction and prepares non-stabilizer states-allowing advantage even when entanglement is lower than in Haar-random states. Moreover, due to all-to-all connectivity in the target distribution, comparable tensor-network simulations demand rapidly growing bond dimensions at fixed error, whereas quantum circuits avoid entanglement truncation and remain a natural tool for sampling from such models.

REFERENCES

[1] N. Bhatnagar, A. Bogdanov, and E. Mossel, in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 14th International Workshop, APPROX 2011, and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings* (Springer, 2011) pp. 424–435.
[2] D. J. Hsu, A. Kontorovich, and C. Szepesvári, Advances in neural information processing systems **28** (2015).
[3] Y. F. Atchadé, SIAM Journal on Mathematics of Data Science **3**, 854 (2021).
[4] D. Layden, G. Mazzola, R. V. Mishmash, M. Motta, P. Wocjan, J.-S. Kim, and S. Sheldon, Nature **619**, 282 (2023).
[5] J.-S. Wang and R. H. Swendsen, Physica A: Statistical Mechanics and its Applications **167**, 565 (1990).
[6] E. Luijten, in *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1* (Springer, 2006) pp. 13–38.
[7] U. Wolff, Physical Review Letters **62**, 361 (1989).
[8] W. Krauth, arXiv preprint cond-mat/0311623 (2003).
[9] J.-C. Walter and G. T. Barkema, Physica A: Statistical Mechanics and its Applications **418**, 78 (2015).
[10] K. Zyczkowski and H.-J. Sommers, Journal of Physics A: Mathematical and General **33**, 2045 (2000).
[11] G. Alagic, C. Majenz, and A. Russell, in *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part III 39* (Springer, 2020) pp. 759–787.
[12] G. O. Roberts and R. L. Tweedie, Bernoulli **2**, 341 (1996).
[13] A. Brown and G. L. Jones, Wiley Interdisciplinary Reviews: Computational Statistics **16**, e70002 (2024).
[14] F. Liang, C. Liu, and R. Carroll, *Advanced Markov chain Monte Carlo methods: learning from past samples* (John Wiley & Sons, 2011).
[15] M. Biron-Lattes, N. Surjanovic, S. Syed, T. Campbell, and A. Bouchard-Côté, in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2024) pp. 4600–4608.
[16] Z. Trstanova, *Mathematical and algorithmic analysis of modified Langevin dynamics*, Ph.D. thesis, Université Grenoble Alpes (2016).
[17] G. O. Roberts and O. Stramer, Methodology and computing in applied probability **4**, 337 (2002).

[18] M. Betancourt, arXiv preprint arXiv:1701.02434 (2017).
[19] M. Betancourt, Annalen der Physik **531**, 1700214 (2019).
[20] R. M. Neal *et al.*, Handbook of markov chain monte carlo **2**, 2 (2011).
[21] B. Ghojogh, H. Nekoei, A. Ghojogh, F. Karray, and M. Crowley, arXiv preprint arXiv:2011.00901 (2020).
[22] Y. Chen and K. Gatmiry, arXiv preprint arXiv:2304.04724 (2023).
[23] X. Tan, arXiv preprint arXiv:2407.15808 (2024).
[24] A. Ransford, M. S. Allman, J. Arkinstall, J. P. C. III, S. F. Cooper, R. D. Delaney, J. M. Dreiling, B. Estey, C. Figgatt, A. Hall, A. A. Husain, A. Isanaka, C. J. Kennedy, N. Kotibhaskar, I. S. Madjarov, K. Mayer, A. R. Milne, A. J. Park, A. P. Reed, R. Ancona, M. P. Andersen, P. Andres-Martinez, W. Angenent, L. Argueta, B. Arkin, L. Ascarrunz, W. Baker, C. Barnes, J. Bartolotta, J. Berg, R. Besand, B. Bjork, M. Blain, P. Blanchard, R. Blume-Kohout, M. Bohn, A. Borgna, D. Y. Botamanenko, R. Boutelle, N. Brown, G. T. Buckingham, N. Q. Burdick, W. C. Burton, V. Carey, C. J. Carron, J. Chambers, J. Children, V. E. Colussi, S. Crepinsek, A. Cureton, J. Davies, D. Davis, M. DeCross, D. Deen, C. Delaney, D. DelVento, B. J. DeSalvo, J. Dominy, R. Duncan, V. Eccles, A. Edgington, N. Erickson, S. Erickson, C. T. Ertsgaard, B. Evans, T. Evans, M. I. Fabrikant, A. Fischer, C. Foltz, M. Foss-Feig, D. Francois, B. Freyberg, C. Gao, R. Garay, J. Garvin, D. M. Gaudiosi, C. N. Gilbreth, J. Giles, E. Glynn, J. Graves, A. Hansen, D. Hayes, L. Heidemann, B. Higashi, T. Hilbun, J. Hines, A. Hlavaty, K. Hoffman, I. M. Hoffman, C. Holliman, I. Hooper, B. Horning, J. Hostetter, D. Hothem, J. Houlton, J. Hout, R. Hutson, R. T. Jacobs, T. Jacobs, M. Johannsen, J. Johansen, L. Jones, S. Julian, R. Jung, A. Keay, T. Klein, M. Koch, R. Kondo, C. Kong, A. Kosto, A. Lawrence, D. Liefer, M. Lollie, D. Lucchetti, N. K. Lysne, C. Lytle, C. MacPherson, A. Malm, S. Mather, B. Mathewson, D. Maxwell, L. McCaffrey, H. McDougall, R. Mendoza, M. Mills, R. Morrison, L. Narmour, N. Nguyen, L. Nugent, S. Olson, D. Ouellette, J. Parks, Z. Peters, J. Petricka, J. M. Pino, F. Polito, M. Preidl, G. Price, T. Proctor, M. Pugh, N. Ratcliff, D. Raymondson, P. Rhodes, C. Roman, C. Roy, C. Ryan-Anderson, F. B. Sanchez, G. Sangiolo, T. Sawadski, A. Schaffer, P. Schow, J. Sedlacek, H. Semenenko, P. Shevchuk, S. Shore, P. Siegfried, K. Singhal, S. Sivarajah, T. Skripka, L. Sletten, B. Spaun, R. T. Sprenkle, P. Stoufer, M. Tader, S. F. Taylor, T. H. Thompson, R. Tobey, A. Tran, T. Tran, G. Vittorini, C. Volin, J. Walker, S. White, D. Wilson, Q. Wolf, C. Wringe, K. Young, J. Zheng, K. Zuraski, C. H. Baldwin, A. Chernoguzov, J. P. Gaebler, S. J. Sanders, B. Neyenhuis, R. Stutz, and J. G. Bohnet, Helios: A 98-qubit trapped-ion quantum computer (2025), arXiv:2511.05465 [quant-ph].
[25] R. Doig and L. Wang, A unified framework for multiple-try metropolis algorithms (2025), arXiv:2503.11583 [stat.CO].
[26] M. B. Lykkegaard, T. J. Dodwell, C. Fox, G. Mingas, and R. Scheichl, Multilevel delayed acceptance mcmc (2022), arXiv:2202.03876 [stat.ME].
[27] J. Diestel and A. Spalsbury, *The joys of Haar measure* (American Mathematical Soc., 2014).
[28] C. Spengler, M. Huber, and B. C. Hiesmayr, Journal of mathematical physics **53** (2012).
[29] A. A. Mele, Quantum **8**, 1340 (2024).
[30] M. Howes, Haar distributed random matrices, Lecture notes or seminar slides (2024), presented at the Student Probability Seminar, March 8, 2024.
[31] B. Kraft and G. Minton, Entries of random matrices (2010), undergraduate research paper, Carleton College.
[32] J. Lin, Department of Mathematics and Statistics, Queens University **40** (2016).
[33] K. W. Ng, G.-L. Tian, and M.-L. Tang, *Dirichlet and Related Distributions: Theory, Methods and Applications* (John Wiley & Sons, Hoboken, NJ, 2011).
[34] C. Walck *et al.*, *Hand-book on statistical distributions for experimentalists* (Stockholms universitet, 1996).
[35] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions* (John Wiley & Sons, 2011).
[36] M. Ledoux, *The concentration of measure phenomenon*, 89 (American Mathematical Soc., 2001).
[37] M. Avalle and A. Serafini, arXiv preprint arXiv:2308.15463 (2023).
[38] Z.-W. Liu, S. Lloyd, E. Zhu, and H. Zhu, Journal of High Energy Physics **2018**, 1 (2018).
[39] S. Sen, Physical review letters **77**, 1 (1996).
[40] S. Aaronson, Nature Physics **11**, 291 (2015).
[41] J. Van Gael, Semantic Scholar (2005).
[42] G. Smith and D. Leung, Physical Review A—Atomic, Molecular, and Optical Physics **74**, 062314 (2006).
[43] V. Sharma and E. J. Mueller, Physical Review A **112**, 012411 (2025).
[44] D. Gottesman, arXiv preprint quant-ph/9807006 (1998).
[45] M. Nest, arXiv preprint arXiv:0811.0898 (2008).
[46] X. Zhang, Z. Pan, and G. Liu, Nature Communications **15**, 10513 (2024).
[47] S. Bravyi, D. Gosset, and R. König, Science **362**, 308 (2018).
[48] A. B. Watts and N. Parham, arXiv preprint arXiv:2301.00995 (2023).
[49] M. Yoganathan, R. Jozsa, and S. Strelchuk, Proceedings of the Royal Society A **475**, 20180427 (2019).