Walking the Hypercube: Unbiased Quantum Partition Functions Without the Matrix

Avinava Dubey[*1] **Arijit Sehanobish**[*2] **Krzysztof Choromanski**[*3,4]

[1]Google Research [2]Kensho Technologies [3]Google DeepMind [4]Columbia University

Abstract

Computing the partition function $\mathcal{Z} = \mathrm{tr}(e^{-\beta H})$ of a quantum spin Hamiltonian H on n sites is a fundamental problem in statistical physics and quantum chemistry. However exact evaluation requires $O(8^n)$ time and $O(4^n)$ memory making it intractable for all but the smallest systems. We propose an unbiased stochastic estimator of $\mathcal{Z}$, and more generally of $\mathrm{tr}(f(H))$ for any function f admitting a convergent power series over exponentially large, implicitly defined operators using only local random walks, avoiding both matrix materialization and matrix-vector products. Our approach combines Graph Random Features with the Hutchinson's stochastic trace estimator, and exploits the algebraic structure of quantum spin Hamiltonians to reduce the per-step computational cost to $O(n + d)$, where d is the mean degree of the spin coupling graph, independent of the Hilbert space dimension $N = 2^n$. We validate the estimator empirically against exact matrix exponentiation for small system sizes, and demonstrate its applicability to system sizes where exact methods are infeasible.

1 Introduction

Computing traces of matrix functions of the form $\mathrm{tr}(f(H))$, where H is a Hamiltonian matrix, is a central computational primitive across statistical physics, quantum chemistry, and machine learning. In statistical physics, the partition function $\mathcal{Z} = \mathrm{tr}(e^{-\beta H})$ at inverse temperature $\beta > 0$ encodes all thermodynamic properties of a quantum system. Similarly, in quantum chemistry, traces of resolvent operators yield density of states estimates, while in machine learning, traces of matrix exponentials appear in graph kernels [4] and Gaussian process inference on graphs [7].

Although evaluating $\mathrm{tr}(f(H))$ is fundamental across various fields, computing it exactly is computationally intractable. For a quantum spin system on n sites, H acts on a Hilbert space of dimension $N = 2^n$. Forming H explicitly requires $\mathcal{O}(4^n)$ memory, and computing $\mathrm{tr}(e^{-\beta H})$ via diagonalization costs $\mathcal{O}(8^n)$ time. Even approximating the partition function of a classical Ising model is #P-hard in general [11], and the quantum case is no easier. Existing practical approaches such as quantum Monte Carlo (QMC) methods [20], tensor network methods [23], and stochastic Lanczos quadrature [22] all require problem-specific structure, introduce approximation errors, or scale poorly with system size.

In this work, we propose a a fully local stochastic framework for estimating tr(f(H)) where H is defined implicitly over an exponential state space for a broad class of quantum spin Hamiltonians, without ever materializing the matrix or performing matrix-vector products. Our approach leads to a per-step computational cost that is polynomial in n and strictly independent of $N = 2^n$. While the total number of samples required may scale exponentially due to the fundamental #P-hardness of the problem, our approach strictly decouples space complexity from time complexity. We reduce the memory footprint to $\mathcal{O}(n)$ and the per-step time to $\mathcal{O}(n + d)$. Our methodology combines two

[*]Equal Contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

existing tools: (1) Graph Random Features (GRF) [4, 18], and (2) Hutchinson's stochastic trace estimator [10]. Unlike previous works on GRFs that require an explicit materialization of H, we bypass direct construction entirely by exploiting the specific algebraic and graph-theoretic structure of each Hamiltonian class to reduce the per-step cost as far as possible.

Setting and Models. We consider Hamiltonians of the form $H = D + M$, where D is a diagonal operator encoding spin-spin interactions and M is an off-diagonal operator encoding a transverse field. The key structural property we exploit is that M induces a sparse, structured graph on the N-dimensional configuration space $\{-1, +1\}^n$. For the models considered in this work, this corresponds to the n-dimensional hypercube Q_n, where neighborhoods of any vertex can be enumerated in $\mathcal{O}(n)$ time via bit operations, without storing the $N \times N$ matrix. The diagonal D is maintained incrementally across walk steps at a cost of $\mathcal{O}(d)$ per step, where d is the mean degree of the coupling graph induced by the spin-spin interaction matrix J. We develop and analyze our estimator for three Hamiltonian classes of physical and theoretical interest:

1. **Transverse Field Ising Model (TFIM).** $H = -\sum_{i<j} J_{ij} Z_i Z_j - \sum_i h_i X_i$, where J_{ij} are pairwise spin couplings and $h_i > 0$ are transverse field strengths. This is the canonical model of quantum phase transitions [19]. The off-diagonal part $M = -\sum_i h_i X_i$ induces the hypercube graph Q_n with edge weights $-h_i$. The coupling graph of J has degree d, giving a per-step cost of $\mathcal{O}(n + d)$. For sparse couplings (e.g. 1D chain, 2D grid), $d = \mathcal{O}(1)$ and the cost reduces to $\mathcal{O}(n)$.
2. **Sherrington-Kirkpatrick (SK) Model.** $H = -\sum_{i<j} J_{ij} Z_i Z_j - \sum_i h_i X_i$, where $J_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, J^2/n)$ are random couplings. This is the canonical spin glass model [21]. The coupling graph is dense ($d = n - 1$), so the incremental diagonal update costs $\mathcal{O}(n)$ per step, which is optimal since any spin flip necessarily changes all $n - 1$ local fields.
3. **XXZ Model.** $H = \sum_{i<j} J_{ij}(X_i X_j + Y_i Y_j) + \Delta \sum_{i<j} J_{ij} Z_i Z_j$, where Δ is an anisotropy parameter ($\Delta = 1$: Heisenberg model, $\Delta = 0$: XX model). The off-diagonal part M now generates two-spin swap moves that preserve total magnetization, giving each configuration $\mathcal{O}(n^2)$ neighbors. The walk graph is no longer the hypercube but remains highly structured. The per-step cost is $\mathcal{O}(n^2)$ for neighbor generation and $\mathcal{O}(d_i + d_j)$ for the incremental diagonal update, where i, j are the two flipped spins.

Contributions. Our main contribution is the following: For each of the above models and any function f admitting a convergent power series $f(H) = \sum_{k=0}^{\infty} \alpha_k H^k$, we construct an estimator $\hat{\mathcal{Z}}$ of $\mathcal{Z} = \operatorname{tr}(f(H))$ that guarantees the following:

- **unbiasedness and sample complexity:** $\mathbb{E}[\hat{\mathcal{Z}}] = \operatorname{tr}(f(H))$, with absolutely no approximation of H or f. The number of samples M for (ϵ, δ)-accuracy scales as $\exp(O(\beta n d/\sqrt{1-p}))$, with no dependence on $N = 2^n$ (Theorem 3.1). Empirically, the relative variance remains stable across system sizes at moderate temperatures.
- **Polynomial per-step cost:** Each random walk step costs $\mathcal{O}(n + d)$ for the TFIM and SK models. This cost has no dependence on $N = 2^n$.
- **No matrix storage:** The $N \times N$ matrix H is never explicitly formed. All quantities, including neighbors, edge weights, and diagonal energies, are computed on-the-fly from the $\mathcal{O}(n^2)$ parameter matrices J and the $\mathcal{O}(n)$ vector h. We demonstrate that HWE matches exact diagonalization to within a few percent for small n, scales to $n = 50$ ($2^{50} \approx 10^{15}$ dimensional Hilbert space) with $\sim$620 MB memory, and produces unbiased estimates on frustrated systems where QMC fails. Since each Hutchinson sample is independent, the computation is embarrassingly parallel, achieving $25\times$ speedup on 32 workers.

The efficiency of our estimator stems from the observation that the graph structure of H, specifically the hypercube structure of M and the incremental update-ability of D allows both the GRF walks and the Hutchinson matrix-vector products to be executed dynamically without ever materializing H. This approach successfully synthesizes GRF [4], which provides an unbiased estimator of any entry of $f(H)$ via random walks on the graph of H, with Hutchinson's estimator [10], which converts matrix-vector products $f(H)z$ into a trace estimate via random Rademacher vectors $z \in \{-1, +1\}^N$.

Organization. Section 3.1 reviews the GRF and Hutchinson frameworks. Sections 3.2, 3.3 and 3.4 present the algorithm and its per-step cost analysis for each model. Theorem 3.3 in Section 3.1 proves exact unbiasedness. Section 4 reports empirical comparisons against exact computation via matrix exponentiation for small n, alongside several well-established baselines, and demonstrates scaling behavior for large n where exact methods are infeasible.

2 Related Work

Stochastic trace estimation. Estimating $\mathrm{tr}(A)$ without explicitly forming A dates back to the seminal work of Hutchinson [10], who introduced the estimator $\mathrm{tr}(A) \approx \frac{1}{S}\sum_s z_s^\top A z_s$ for random vectors z_s. [2] provided the first rigorous sample complexity bounds for several distributions z, showing that Rademacher vectors achieve minimum variance while Gaussian vectors yield better concentration bounds. Moreover there has been several works [14, 12] that improve the Hutchinson's algorithm. Other extensions to Hutchinson's algorithm include matrix functions, where [22] combined Hutchinson's estimator with the Lanczos algorithm and Gaussian quadrature to estimate $\mathrm{tr}(f(A))$ for analytic f and symmetric positive definite A, with provable error bounds. [9] instead utilized stochastic Chebyshev approximations for the same problem. All these methods require matrix-vector products with A, which in our setting imposes an intractable cost of $\mathcal{O}(N^2) = \mathcal{O}(4^n)$ per product if H is formed explicitly. Our framework avoids this exponential overhead, reducing the per-step cost to $\mathcal{O}(n + d)$ by exploiting the algebraic structure of the system directly.

Quantum Monte Carlo (QMC). QMC methods is an important baseline for simulating quantum spin systems at finite temperature. [20] introduced the Stochastic Series Expansion (SSE) method, which represents the partition function $\mathcal{Z} = \mathrm{tr}(e^{-\beta H})$ as a power series in βH and samples the resulting discrete configuration space via Markov chain Monte Carlo.

Tensor network methods. Tensor network approaches, primarily matrix product states (MPS) and the density matrix renormalization group (DMRG), provide efficient representations of quantum states for systems with limited entanglement [25, 23]. Estimating $\mathrm{tr}(e^{-\beta H})$ via tensor networks typically involves imaginary-time evolution, which inherently introduces Trotter errors by splitting $e^{-\beta H}$ into local factors. While computationally efficient for 1D systems, the cost scales exponentially in higher dimensions.

Kernel methods on graphs and Graph Random Features (GRF). Graph kernels based on matrix functions of the adjacency or Laplacian matrix,including the diffusion kernel [13], the p-step random walk kernel, and the regularized Laplacian kernel are foundational to graph classification and regression tasks in machine learning [24]. Computing these kernels exactly requires $\mathcal{O}(N^3)$ operations. This has motivated scalable algorithms for the computation of the action of exponential of a matrix [1, 3, 16, 15]. While these methods only work for the matrix exponentials, [4] introduced GRFs to construct unbiased estimators of certain graph kernels of the form $f(A)$ via random walks on the graph induced by A. The main insight is that the power series expansion $f(A) = \sum_k \alpha_k A^k$ can be estimated entry-by-entry by sampling random walks of length k with probability proportional to the absolute values of the edge weights, subsequently correcting for the sampling distribution via importance weighting. The computational efficiency and attractive theoretical properties of GRFs have spurred extensive follow-up work optimizing the mechanism [17, 18, 8, 6, 5, 26]. However, a limitation of these existing works on GRFs requires *explicit materialization and storage of the adjacency or Laplacian matrix*, which is computationally intractable for quantum systems. To the best of our knowledge, this work represents the first application of GRFs to quantum Hamiltonians and our contribution can be interpreted as extending graph kernel approximations to the exponentially large configuration space of a quantum spin system. Here, the graph structure of H, specifically the hypercube topology of the transverse field M and the sparse coupling graph of J enables on-the-fly computation of all walk quantities at a cost of $\mathcal{O}(n + d)$ per step, strictly independent of the Hilbert space dimension $N = 2^n$. This enables us to bypass the naive GRF algorithm which would be prohibitively expensive in this case.

Hardness of partition function computation. Exact computation of the partition function $\mathcal{Z} = \mathrm{tr}(e^{-\beta H})$ is strictly #P-hard in the general case. [11] demonstrated that the partition function of the classical ferromagnetic Ising model admits a fully polynomial randomized approximation scheme

(FPRAS), but the problem becomes #P-hard when general (frustrated) couplings are introduced. The quantum variant considered here is at least as computationally demanding as the classical case: setting $h_i = 0$ for all i reduces H to the classical Ising Hamiltonian $H = -\sum_{i<j} J_{ij} Z_i Z_j$, whose partition function is exactly the classical Ising partition function. Our estimator does not circumvent these fundamental hardness results; rather, it isolates the exponential bottleneck entirely within the variance of the Monte Carlo sampling. Unlike exact methods that fail deterministically due to $\mathcal{O}(4^n)$ memory exhaustion, our framework trades intractable space complexity for sample complexity.

3 Walking the Hypercube

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a state-space graph on $N = 2^n$ vertices, where each vertex $x \in \{-1, +1\}^n$ corresponds to a spin configuration of an n-spin system in the computational basis. We investigate the thermodynamic properties of these systems by estimating the partition function trace at inverse temperature $\beta > 0$:

$$\mathcal{Z} = \mathrm{tr}\big(e^{-\beta H}\big), \tag{1}$$

where $H \in \mathbb{R}^{N \times N}$ is the system's Hamiltonian. Evaluating Eq. 1 is classically intractable for large n due to the necessity of computing the matrix exponential of an exponentially large operator.

In this section, we present a generalized zero-memory stochastic estimator for $\mathcal{Z}$ that bypasses this $\mathcal{O}(4^n)$ memory bottleneck. We first establish the framework combining Graph Random Features (GRFs) and Hutchinson's trace estimator (Section 3.1). We then detail how this unified algorithm adapts dynamically to the specific graph-theoretic and algebraic structures of three distinct quantum systems without ever materializing the underlying matrices (Sections 3.2 - 3.4).

3.1 Hypercube Walk Estimator (HWE)

To bypass explicitly forming H or $e^{-\beta H}$, we leverage Graph Random Features (GRFs). Following [4, 18], we define the GRF mechanism below.

Let H be a weighted graph (with possibly negative weights) with a set of vertices $\mathcal{V}$. Define $\mathcal{W}(v)$ as the distribution over random walks starting at vertex v, where at each step the walk moves from vertex u to a neighbor w with probability proportional to $|H_{uw}|$, and halts at each step independently with probability $p \in (0, 1)$.

To establish a rigorous theoretical guarantee on sample complexity, we first consider a *Vanilla* walk variant. For a walk $\omega = (v_0, v_1, \ldots, v_L)$ of length L, this variant drops a weight only at the final node:

$$\hat{\phi}_{\mathrm{RR}}(\omega)_w = \frac{\alpha_L}{p(1-p)^L} \left(\prod_{s=0}^{L-1} \frac{H_{v_s v_{s+1}}}{t_{v_s v_{s+1}}} \right) \mathbb{I}[v_L = w] \tag{2}$$

where $t_{uv} = |H_{uv}| / \sum_w |H_{uw}|$ is the transition probability. By eliminating mathematical cross-terms between steps, this formulation allows us to bound the estimator's variance:

Theorem 3.1 (Sample Complexity)**.** *To estimate $\mathcal{Z}$ within relative error ϵ with probability $1 - \delta$, the Vanilla HWE requires $M = \mathcal{O}\left(\frac{1}{\epsilon^2 \delta} \exp\left(\mathcal{O}\left(\frac{\beta n d}{\sqrt{1-p}}\right)\right)\right)$ samples. (Proof in Appendix).*

While theoretically tractable, discarding intermediate walk steps wastes valuable data and increases empirical variance. To maximize the signal extracted from every trajectory in practice, our implementation utilizes a *Continuous Trail* walk. Instead of scoring only the final node, we define the continuous random feature vector by accumulating unbiased contributions at every step ℓ:

$$\hat{\phi}(\omega)_w = \sum_{\ell=0}^{L} \alpha_\ell \frac{\prod_{s=0}^{\ell-1} H_{v_s v_{s+1}}}{\prod_{s=0}^{\ell-1} (1-p) \cdot t_{v_s v_{s+1}}} \mathbb{I}[v_\ell = w] \tag{3}$$

The GRF estimator

$$\hat{\Phi}(v) = \frac{1}{R} \sum_{r=1}^{R} \hat{\phi}(\omega^{(r)}), \quad \omega^{(r)} \overset{\text{i.i.d.}}{\sim} \mathcal{W}(v). \tag{4}$$

is provably unbiased for the v-th column of $g(H)$:

Proposition 3.2 (Unbiasedness of GRF, [4]). *For any $v, w \in V$,*

$$\mathbb{E}_{\mathcal{W}(v)}\left[\hat{\Phi}(v)_w\right] = [g(H)]_{vw}. \tag{5}$$

We decompose $f(H) = g(H)^2$, where $g(H) = e^{-\frac{\beta}{2}H}$. We fuse this decomposition with Hutchinson's stochastic trace estimator [10]. Recall that for any symmetric matrix and a random vector $z \in \{-1, +1\}^N$ with i.i.d. Rademacher entries, $\mathbb{E}_z\left[z^\top A z\right] = \mathrm{tr}(A)$. By simulating two independent random walks $\omega^{(1)}$ and $\omega^{(2)}$ from v to estimate $g(H)$, we define the Two-Walk GRF-Hutchinson estimator:

$$\hat{\mathcal{Z}} = \frac{1}{S}\sum_{s=1}^{S}\sum_{v\in\mathcal{V}}\frac{1}{R}\sum_{r=1}^{R}\underbrace{\left(z^{(s)\top}\hat{\phi}(\omega^{(1,r)})\right)}_{\text{Walk A Sum}}\underbrace{\left(z^{(s)\top}\hat{\phi}(\omega^{(2,r)})\right)}_{\text{Walk B Sum}}. \tag{6}$$

Algorithm 1 outlines the procedural implementation of the HWE, operationalizing the Two-Walk GRF-Hutchinson estimator defined in Equation 4. To compute the partition function $\mathcal{Z} = \mathrm{tr}(e^{-\beta H})$ without ever materializing the exponentially large Hamiltonian H, the algorithm exploits the decomposition $f(H) = g(H)^2$ alongside Hutchinson's trace estimator. For a chosen number of samples S, independent Rademacher probe vectors $z^{(s)}$ are drawn. The algorithm then iterates over sampled root vertices $v \in \mathcal{V}$, simulating R independent pairs of random walks ($\omega^{(1)}$ and $\omega^{(2)}$) from each vertex. The random features generated by these walks, $\hat{\phi}(\omega)$, act as unbiased estimators for the v-th column of $g(H) = e^{-\frac{\beta}{2}H}$. By projecting these feature vectors onto the probe vector $z^{(s)}$, we efficiently compute two independent scalar values (Walk A Sum and Walk B Sum). Accumulating the product of these paired sums and averaging across all walk pairs, root vertices, and probe vectors yields an unbiased, zero-memory stochastic estimate of the target trace $\hat{\mathcal{Z}}$.

Theorem 3.3 (Unbiasedness of Two-Walk HWE). *The estimator $\hat{\mathcal{Z}}$ satisfies $\mathbb{E}\left[\hat{\mathcal{Z}}\right] = \mathrm{tr}(f(H))$.*

Proof. By the tower property and independence of the walks and z: $\mathbb{E}\left[\hat{\mathcal{Z}}\right] = \sum_v \mathbb{E}_z\left[z^\top \mathbb{E}[\hat{\phi}^{(1)}]\mathbb{E}[\hat{\phi}^{(2)}]^\top z\right]$. Since $\mathbb{E}[\hat{\phi}] = g(H)_{:v}$, this evaluates to $\sum_v \mathbb{E}_z\left[z^\top g(H)_{:v} g(H)_{:v}^\top z\right]$. By Hutchinson's property, this is $\sum_v \langle g(H)_{:v}, g(H)_{:v}\rangle = \sum_v [g(H)^2]_{vv} = \mathrm{tr}(f(H))$. □

Algorithm 1 Hypercube Walk Estimator (HWE)

Objective: Dynamic trace $\mathrm{tr}(f(H))$ estimation without the exponentially large Hamiltonian H.

Draw S independent probe vectors $z^{(s)} \sim \mathrm{Rad}(N)$.
for sampled root states $v \in \mathcal{V}$ **do**
  for walk pair $r = 1 \dots R$ **do**
    Walk A: Simulate continuous walk $\omega^{(1)}$ from v.
    Compute Hutchinson Sum A: $P_1 = z^{(s)\top}\hat{\phi}(\omega^{(1)})$
    Walk B: Simulate continuous walk $\omega^{(2)}$ from v.
    Compute Hutchinson Sum B: $P_2 = z^{(s)\top}\hat{\phi}(\omega^{(2)})$
    Accumulate inner product: $\hat{Z}_v \leftarrow \hat{Z}_v + (P_1 \times P_2)$
return $\hat{\mathcal{Z}} = \frac{1}{S}\sum_{s=1}^{S}\sum_{v\in\mathcal{V}}\frac{\hat{Z}_v}{R}$ ▷ Unbiased Trace Estimate

The resulting algorithm achieves a theoretical total complexity of $\mathcal{O}(M \cdot S \cdot R \cdot L \cdot n)$, where M is the number of sampled states.

3.2 Adaptation for TFIM

The TFIM Hamiltonian is:

$$H_{\mathrm{TFIM}} = -\sum_{i<j} J_{ij} Z_i Z_j - \sum_{i=1}^{n} h_i X_i. \tag{7}$$

Decomposing the Hamiltonian into $H = D + M$, the diagonal matrix D captures the classical energy interactions, and the off-diagonal matrix $M = -\sum_i h_i X_i$ encodes the quantum fluctuations.

The core mechanism relies on tracking the graph topology mathematically rather than storing it in memory. The off-diagonal support graph forms an n-dimensional hypercube where the n neighbors of configuration x are computable via bitwise flips ($\mathcal{O}(n)$ time). The diagonal energy $D_{xx} = -\sum_{i<j} J_{ij}x_ix_j$ is never recomputed from scratch. By maintaining a vector of local fields, the energy impact of a spin flip at index i requires updating only the fields of its immediate neighbors. For sparse coupling matrices with mean degree d, this isolates the entire diagonal update strictly to $\mathcal{O}(d_i) = \mathcal{O}(1)$ operations. Precise implementation of HWE walk are defined in Algorithm 2.

Algorithm 2 TFIM Subroutines for HWE

Objective: Execute HWE steps dynamically via $\mathcal{O}(n + d_i)$ state updates.

function INITWALK(v) ▷ Plugs into Alg. 1, Walk A, B
  $x \leftarrow v, \quad \ell_i \leftarrow \sum_{j\in\mathcal{N}(i)} J_{ij}x_j \ (\forall i), \quad D_{xx} \leftarrow -\frac{1}{2}\sum_i x_i\ell_i$
  return x, ℓ, D_{xx}

function TAKESTEP(x, ℓ, D_{xx}) ▷ Plugs into Alg. 1, Walk A, B
  Implicit Weights: $H_{x,x\oplus e_i} \leftarrow -h_i, \quad H_{xx} \leftarrow D_{xx}$
  Transition $x \to x' = x \oplus e_i$ wp $\propto h_i$, or stay wp $\propto |D_{xx}|$
  if walk flipped spin i to transition **then**
    $D_{x'x'} \leftarrow D_{xx} + 2x_i\ell_i, \quad \ell_j \leftarrow \ell_j - 2J_{ij}x_i \ (\forall j \in \mathcal{N}(i)), \quad x_i \leftarrow -x_i$
  return $x, \ell, D_{x'x'}$

3.3 Adaptation for the SK Model

Beyond sparse lattices, the HWE scales to dense systems such as the Sherrington-Kirkpatrick (SK) spin glass model. The SK model features dense, all-to-all random couplings $J_{ij} \sim \mathcal{N}(0, J^2/n)$. Because every spin couples to every other spin, the physical coupling graph degree is $d = n - 1$. Despite this complete connectivity, the neighbor generation cost on the walk graph remains $\mathcal{O}(n)$, identical to the TFIM. The incremental diagonal update defined previously is bounded by $\mathcal{O}(n)$ per step. While every spin flip now necessarily alters all $n - 1$ local fields, this $\mathcal{O}(n)$ update cost is mathematically optimal for dense coupling matrices and eliminates the $\mathcal{O}(n^2)$ overhead of naive energy recalculation. The $\mathcal{O}(n^2)$ initialization and $\mathcal{O}(n)$ dynamic updates are detailed in Algorithm 3 in appendix section B.

3.4 Adaptation for the XXZ Model

To demonstrate that our framework is *not restricted to hypercube walk graphs*, we adapt the execution for the XXZ Hamiltonian. The XXZ model possesses a $U(1)$ symmetry that restricts transitions to specific constant-magnetization sectors of the Hilbert space. Under this physical constraint, the off-diagonal transition operator M no longer generates single bit-flips, but rather two-spin swap moves ($01 \leftrightarrow 10$) that preserve total magnetization. Consequently, the walk graph is no longer an n-dimensional hypercube; instead, each configuration has $\mathcal{O}(n^2)$ valid neighbors.

The dynamic execution is adapted as follows: neighbor generation scales to $\mathcal{O}(n^2)$ to enumerate all valid swaps, while the incremental local field maintenance scheme reduces the diagonal energy update to $\mathcal{O}(d_i + d_j)$ per transition, where i and j are the indices of the swapped spins. If the underlying coupling J is structured (e.g., a nearest-neighbor lattice), $d_i + d_j = \mathcal{O}(1)$, allowing the walk to proceed independent of system size after an initial $\mathcal{O}(n^2)$ state calculation (see Appendix B for exact implementation details).

4 Experiments

Empirically, HWE maintains a bounded memory footprint and stable relative error across system sizes, avoiding the OOM failures and truncation errors of traditional methods.

Experimental Baselines: We benchmark against five numerical methods:

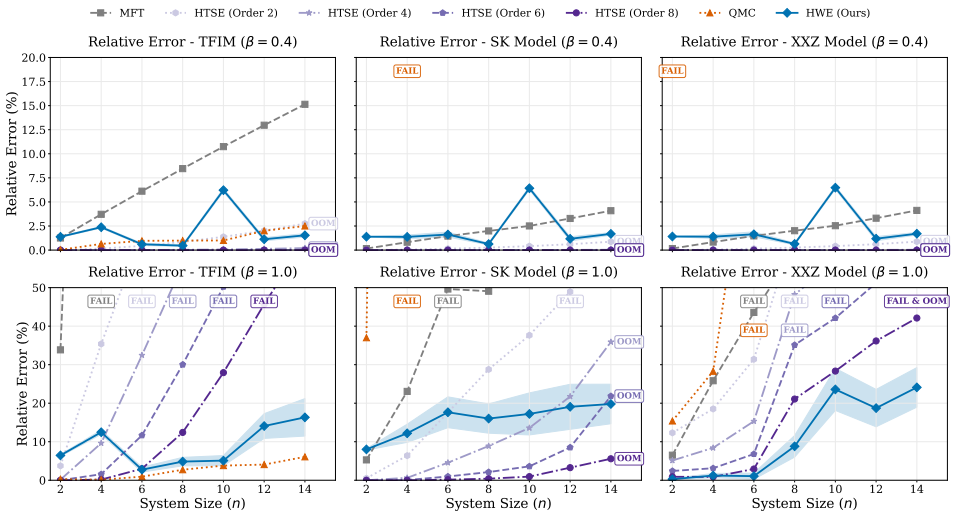


Figure 1: **Relative Error across System Sizes.** A comparison of trace estimators on the TFIM, SK, and XXZ models at high ($\beta = 0.4$, top) and low ($\beta = 1.0$, bottom) temperatures. Methods exceeding a 40% error threshold are strictly labeled as **FAIL**, while those exceeding available memory (32 GB) at $n = 16$ are labeled **OOM**. At low temperatures, MFT fails to capture critical quantum fluctuations, QMC performs well for TFIM but poorly for rest of the evaluations, and HTSE mathematically diverge with increasing n even though they are the most accurate at smaller values of n.

- **Exact Diagonalization (ED):** Computes the full $2^n \times 2^n$ Hamiltonian matrix to solve $H|\psi_k\rangle = E_k|\psi_k\rangle$. The partition function is evaluated exactly as $\mathcal{Z}_{ED}(\beta) = \sum_{k=1}^{2^n} e^{-\beta E_k}$.
- **Mean-Field Theory (MFT):** Approximates the system by assuming each spin interacts with an average effective field. The partition function factorizes to $\mathcal{Z}_{MF}(\beta) = \prod_{i=1}^{n} 2\cosh\left(\beta\sqrt{(\sum_j J_{ij}m_j)^2 + h_i^2}\right)$.
- **Quantum Monte Carlo (QMC):** Implemented via the Stochastic Series Expansion (SSE) variant. The mathematical formulation and mechanism are detailed in Section 2.
- **High-Temperature Series Expansions (HTSE):** Evaluates the partition function via a Taylor series of the matrix exponential: $\mathcal{Z}_{HTSE}(\beta) = \mathrm{tr}(I - \beta H + \frac{\beta^2}{2!}H^2 - \dots)$.

4.1 Standard TFIM

The 1D chain TFIM serves as a sparse benchmark. With a nearest-neighbor coupling matrix J, generating neighbors costs $\mathcal{O}(n)$, and the diagonal energy update costs $\mathcal{O}(d_i)$ per step. Since $d_i = \mathcal{O}(1)$ for a 1D chain, HWE operates at $\mathcal{O}(n)$ per step. As Figure 1 shows, HTSE yield highly accurate estimates for small system sizes and higher temperatures (smaller β). However, the method's truncation error diverges as n grows, a limitation that is compounded as the temperature is lowered ($\beta \to 1.0$). Furthermore, Figure 2 demonstrates that matrix-based methods, including HTSE, exhibit $\mathcal{O}(4^n)$ memory scaling, encountering out-of-memory (OOM) limits before $n = 16$. In contrast, HWE bypasses matrix materialization entirely to maintain a bounded relative error. Memory usage stabilizes around 400-500 MB for larger n, making it straightforward to run simulations up to $n = 20$ and beyond.

4.2 Sherrington-Kirkpatrick Model

The SK model provides a dense, frustrated benchmark, with a Hamiltonian featuring all-to-all random Gaussian couplings, $J_{ij} \sim \mathcal{N}(0, J^2/n)$. With a graph degree of $n - 1$, HWE's incremental diagonal update scales to $\mathcal{O}(n)$ per step.

MFT and QMC performs poorly at low temperatures (Figure 1). Similarly, while HTSE is accurate at high temperatures for small n, its error diverges sharply as n increases and β approaches 1.0. HTSE (order 8) performs better than lower order HTSE methods as expected. Exact matrix methods are

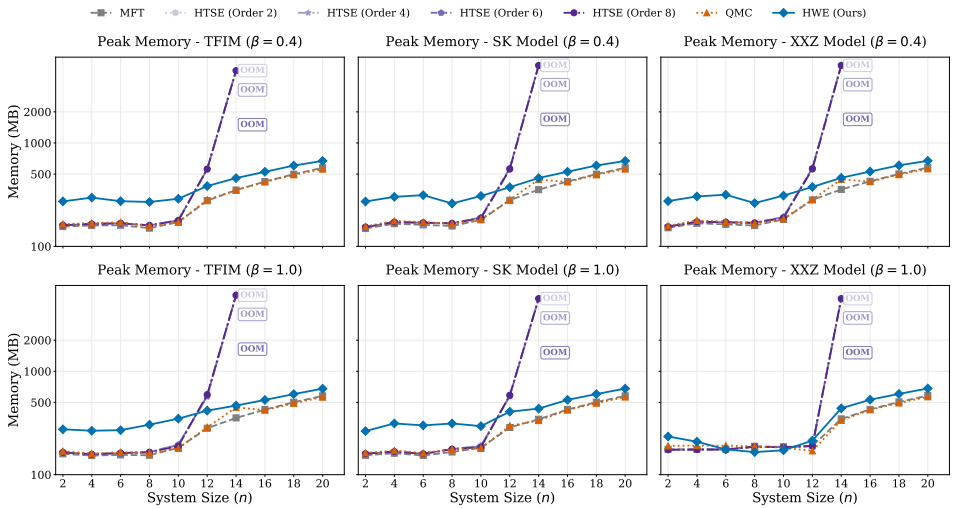


Figure 2: **Peak Memory Scaling.** Memory footprint as a function of system size n. Exact matrix-based methods, including all orders of HTSE, exhibit an expected $O(4^n)$ memory explosion, triggering Out-Of-Memory (OOM, above 32 GB) failures strictly before $n = 16$. In contrast, the GRF estimator computes random walks on-the-fly, entirely bypassing matrix materialization to maintain a bounded memory footprint across all evaluated models.

accurate for small n, but they hit OOM limits by $n = 16$ (Figure 2, memory size of machine 32 GB). HWE yeilds relatively stable errors. As before the method gives the best tradeoff of memory used and accuracy. Consistent with the TFIM results, peak memory remains around 400-500 MB for larger n, allowing the method to scale to $n = 20$.

4.3 XXZ Model

We benchmark the XXZ Hamiltonian at the Heisenberg point ($\Delta = 1.0$) with uniform coupling $J = 0.2$ on a random sparse graph at inverse temperature $\beta = 1.0$. This low temperature amplifies higher-order quantum fluctuations, and the model's $U(1)$ symmetry restricts valid dynamics to constant-magnetization sectors.

As Figure 1 shows, the random graph topology induces frustration, causing QMC to fail. Although HTSE provides good approximations at higher temperatures, the target low-temperature regime ($\beta = 1.0$) severely degrades its accuracy. Meanwhile, exact matrix formation reaches memory limits before $n = 16$ (Figure 2). HWE addresses the symmetry constraint through a magnetization-preserving walk, dynamically generating the $\mathcal{O}(n^2)$ valid two-spin swap transitions rather than explicitly forming the constrained Hilbert space matrix. This yields an unbiased trace estimate while maintaining a consistent memory footprint, scaling to $n = 20$ and requiring roughly 400 MB.

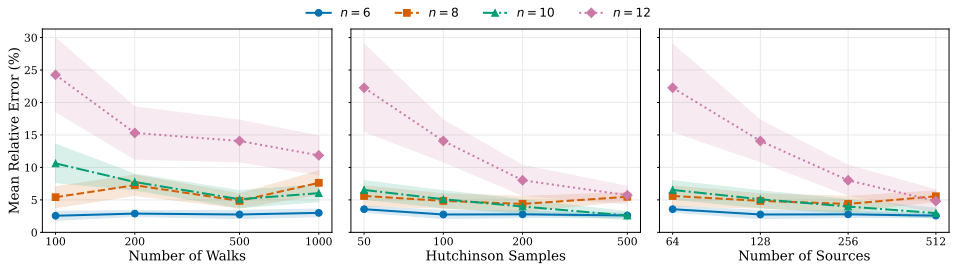


Figure 3: Hyperparameter ablation for HWE. Mean relative error and variance (shaded) decrease monotonically with respect to the number of walks, Hutchinson probes, and source nodes across varying system sizes.

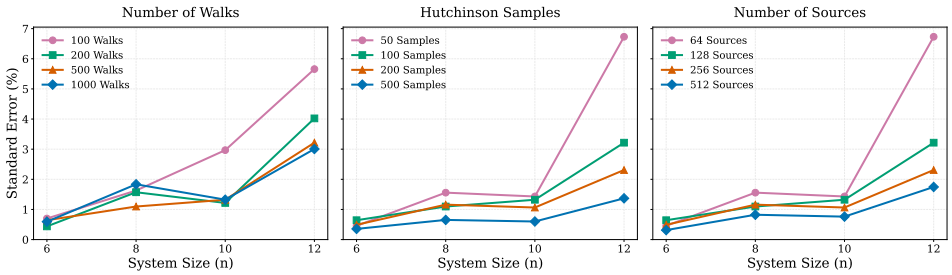


Figure 4: Standard Error scaling of the HWE estimator as a function of system size (n) across varying numbers of Walks, Hutchinson Samples, and Source Nodes.

4.4 Ablation Study

We evaluate the sensitivity of the HWE to its three core sampling hyperparameters: the number of random walks (R), Hutchinson probe vectors (S), and sampled source nodes (M). Figure 3 reports the mean relative error and estimator variance across these dimensions for varying system sizes ($n \in \{6, 8, 10, 12\}$). As expected for a Monte Carlo method, increasing the sample count across any axis monotonically decreases both error and variance. This convergence behavior can be seen across system sizes, confirming that HWE's precision can be tuned independently of the state space dimension N via trivially parallelizable sampling.

Variance Scaling: Figure 4 illustrates the scaling of the estimator's variance with the size of the system n. Consistent with our understanding, the baseline standard error grows due to the problem's underlying #P-hardness. However, increasing the Monte Carlo hyperparameters monotonically suppresses this noise. Because HWE avoids explicit matrix materialization and maintains a $\mathcal{O}(n + d)$ per-step cost, executing the extensive parallel sampling required to bound this variance remains computationally tractable.

4.5 Computation Time

Figure 5 compares the wall-clock execution time for evaluating the TFIM partition function at $n = 14$. ED requires 106 minutes to evaluate the trace. Similarly, the dense matrix multiplications required for HTSE impose substantial computational overhead, resulting in execution times scaling from 66 to 102 minutes as the Taylor expansion order increases. By operating exclusively on dynamically generated local graph transitions and avoiding explicit matrix-vector products, HWE completes the estimation in 49 minutes. While MFT and QMC execute in under one minute, they have some of the worst error (Figure 1).

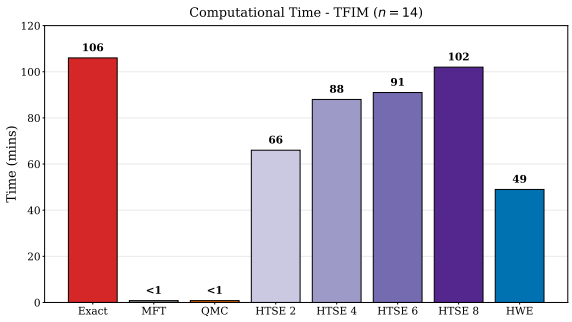


Figure 5: Wall-clock computation time.

4.6 Scalability and Parallelization

Since calculating the partition function for general frustrated Hamiltonians is #P-hard, exact methods inevitably fail due to an $\mathcal{O}(4^n)$ memory limit. HWE relocates this exponential bottleneck from an insurmountable memory barrier to an embarrassingly parallelizable time bottleneck.

Parallel Speedup: Since each Hutchinson sample is strictly independent, the algorithm achieves near-linear scaling with standard parallel compute. Table 1 demonstrates scaling on a TFIM system ($n = 14$), achieving a $25.1\times$ relative speedup on 32 workers, reducing execution from over 15 minutes to under 2 minutes with no degradation in estimator stability.

Table 1: Parallel Speedup for TFIM ($n = 14$).

Workers	Relative Speedup	Rel. Diff in $\hat{\mathcal{Z}}$	$\mathrm{Var}(\hat{\mathcal{Z}})/\hat{\mathcal{Z}}^2$
1	1.00×	—	7×10^{-6}
4	3.62×	-0.98%	1.5×10^{-5}
8	7.78×	+1.90%	1.5×10^{-5}
16	13.72×	-0.16%	1.5×10^{-5}
32	25.12×	-0.16%	1.5×10^{-5}

Large-n Validation: By eliminating the memory barrier, HWE can scale to systems completely unreachable by exact diagonalization. Table 2 shows HWE successfully evaluating a TFIM system up to $n = 50$ (a $2^{50} \approx 10^{15}$ dimensional Hilbert space) using 64 cores. The peak memory remained strictly bounded to ≈ 620 MB.

Table 2: Large-n Scaling for TFIM ($\beta = 0.4$, 64 workers).

n	$\mathrm{Var}(\hat{\mathcal{Z}})/\hat{\mathcal{Z}}^2$	Avg Wall Time	Peak Memory
20	2.0×10^{-5}	371.5s	625.4 MB
30	4.8×10^{-5}	467.8s	623.1 MB
40	2.7×10^{-4}	483.3s	623.3 MB
50	5.8×10^{-3}	508.8s	620.7 MB

To validate accuracy at these massive scales, we utilized two approaches. First, we isolated the off-diagonal transverse field machinery (the $\mathcal{M}$-only estimator, $\hat{\mathcal{Z}}_M = \mathrm{tr}(e^{-\beta M})$), which yielded a stable relative error of 6.79% at $n = 50$. Second, for the full TFIM model, we used the Jordan-Wigner (JW) transformation (which solves 1D unfrustrated chains in $\mathcal{O}(n^3)$ time) as an oracle. Evaluated against JW at $\beta = 0.4$, the GRF estimator's average error scaled gracefully, increasing from 2.5% at $n = 14$ to just 10.25% at $n = 20$.

4.7 The Sign Problem and Variance Scaling

For frustrated quantum systems, stochastic estimators inevitably encounter the sign problem at low temperatures, where rapid sign alterations in the walk paths cause the variance to explode. We quantify this via the effective sign metric $\langle \mathrm{sign} \rangle = \bar{\mathcal{Z}}[\mathrm{signed}]/\bar{\mathcal{Z}}[\mathrm{unsigned}]$. Note that unlike standard Quantum Monte Carlo (QMC), GRF estimates are products of two walk sums containing independent Taylor and Rademacher signs, meaning $\langle \mathrm{sign} \rangle$ can occasionally exceed 1 when constructive interference occurs.

As shown in Table 3 (appendix sec C), the effective sign decays severely at higher β (low temperatures), driving an exponential increase in relative variance. However, at moderate temperatures ($\beta \leq 0.5$), the effective sign decays gracefully.

Crucially, while the relative variance $\mathrm{Var}(\hat{\mathcal{Z}})/\hat{\mathcal{Z}}^2$ is highly sensitive to temperature, it exhibits remarkable stability across scaling system sizes n. At $\beta = 0.4$, the relative variance for the frustrated TFIM remains virtually flat from $n = 4$ (1.0×10^{-4}) to $n = 20$ (1.8×10^{-5}).

5 Conclusion

We introduced HWE, an unbiased $\mathcal{O}(n + d)$ per-step estimator for $\mathrm{tr}(f(H))$ across quantum spin Hamiltonians. By leveraging Graph Random Features and incremental local updates, HWE bypasses explicit matrix formation, matrix-vector products, and Trotter approximations. This provides a strictly unbiased, memory-efficient framework capable of scaling to Hilbert space dimensions where exact matrix methods fail.

References

[1] Awad H Al-Mohy and Nicholas J Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM journal on scientific computing*, 33(2):488–511, 2011.

[2] Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM*, 58:8:1–8:34, 2011.

[3] Philipp Bader, Sergio Blanes, and Fernando Casas. An improved algorithm to compute the exponential of a matrix. *arXiv preprint arXiv:1710.10989*, 2017.

[4] Krzysztof Choromanski. Taming graph kernels with random features. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[5] Krzysztof Choromanski, Avinava Dubey, Arijit Sehanobish, and Isaac Reid. Computationally-efficient graph modeling with refined graph random features. *arXiv preprint arXiv:2510.07716*, 2025.

[6] Krzysztof Choromanski, Isaac Reid, Arijit Sehanobish, and Avinava Dubey. Optimal time complexity algorithms for computing general random walk graph kernels on sparse graphs. *arXiv preprint arXiv:2410.10368*, 2024.

[7] David B Dunson, Hau-Tieng Wu, and Nan Wu. Graph based gaussian processes on restricted domains. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(2):414–439, 2022.

[8] Brooke Feinberg and Aiwen Li. Heating up quasi-monte carlo graph random features: A diffusion kernel perspective. *arXiv preprint arXiv:2410.08389*, 2024.

[9] Insu Han, Dmitry Malioutov, Haim Avron, and Jinwoo Shin. Approximating spectral sums of large-scale matrices using stochastic chebyshev approximations. *SIAM Journal on Scientific Computing*, 39(4):A1558–A1585, 2017.

[10] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communication in Statistics- Simulation and Computation*, 18:1059–1076, 01 1989.

[11] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on Computing*, 22(5):1087–1116, 1993.

[12] Shuli Jiang, Hai Pham, David Woodruff, and Richard Zhang. Optimal sketching for trace estimation. *Advances in Neural Information Processing Systems*, 34:23741–23753, 2021.

[13] Risi Imre Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, page 315–322, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[14] Raphael A Meyer, Cameron Musco, Christopher Musco, and David P Woodruff. Hutch++: Optimal stochastic trace estimation. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 142–155. SIAM, 2021.

[15] Cameron Musco, Christopher Musco, and Aaron Sidford. Stability of the lanczos method for matrix function approximation. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1605–1624. SIAM, 2018.

[16] Lorenzo Orecchia, Sushant Sachdeva, and Nisheeth K Vishnoi. Approximating the exponential, the lanczos method and an o (m)-time spectral algorithm for balanced separator. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1141–1160, 2012.

[17] Isaac Reid, Krzysztof M Choromanski, and Adrian Weller. Quasi-monte carlo graph random features. *Advances in Neural Information Processing Systems*, 36:14770–14796, 2023.

[18] Isaac Reid, Krzysztof Marcin Choromanski, Eli Berger, and Adrian Weller. General graph random features. In *The Twelfth International Conference on Learning Representations*, 2024.

[19] Subir Sachdev. *Quantum Phase Transitions of Antiferromagnets and the Cuprate Superconductors*, page 1–51. Springer Berlin Heidelberg, 2012.

[20] Anders W. Sandvik. Stochastic series expansion method with operator-loop update. *Phys. Rev. B*, 59:R14157–R14160, Jun 1999.

[21] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Phys. Rev. Lett.*, 35:1792–1796, Dec 1975.

[22] Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of $tr(f(a))$ via stochastic lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017.

[23] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical review letters*, 91(14):147902, 2003.

[24] S.V.N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(40):1201–1242, 2010.

[25] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992.

[26] Matthew Zhang, Jihao Andreas Lin, Krzysztof Choromanski, Adrian Weller, Richard E Turner, and Isaac Reid. Graph random features for scalable gaussian processes. *arXiv preprint arXiv:2509.03691*, 2025.

A Proof Theorem 3.1

Proof. Let $\hat{Z}_v = \langle \hat{\phi}^{(1)}, \hat{\phi}^{(2)} \rangle$ be the single-node unbiased trace estimator derived from two statistically independent random walks, such that $\mathbb{E}[\hat{Z}_v] = [f(H)]_{vv}$. Our goal is to bound the sample complexity M required to estimate the total partition function $\mathcal{Z}$ within a relative error ϵ with probability $1 - \delta$.

We first bound the variance of the estimator, which is strictly upper-bounded by its second moment: $\mathrm{Var}(\hat{Z}_v) \leq \mathbb{E}[\hat{Z}_v^2]$. Applying the Cauchy-Schwarz inequality and leveraging the statistical independence of the two walks, we obtain:

$$\mathbb{E}[\hat{Z}_v^2] = \mathbb{E}\left[\langle \hat{\phi}^{(1)}, \hat{\phi}^{(2)} \rangle^2\right] \leq \mathbb{E}\left[\|\hat{\phi}^{(1)}\|_2^2 \|\hat{\phi}^{(2)}\|_2^2\right] = \left(\mathbb{E}\left[\|\hat{\phi}\|_2^2\right]\right)^2. \tag{8}$$

To eliminate cross-terms, we utilize the Vanilla estimator variant, which terminates and drops a single weight at a randomized length L drawn from a geometric distribution with halting probability p. The feature vector is strictly one-hot, yielding a squared L_2 norm of:

$$\|\hat{\phi}\|_2^2 = \frac{\alpha_L^2}{p^2(1-p)^{2L}} \left(\prod_{s=0}^{L-1} \frac{H_{v_s v_{s+1}}}{t_{v_s v_{s+1}}}\right)^2. \tag{9}$$

By definition of the transition probabilities $t_{uv} = |H_{uv}|/\|H_{u:}\|_1$, the edge weight fractions boundedly reduce to the absolute row sum of the current node: $|H_{uv}|/t_{uv} = \|H_{u:}\|_1 \leq W_{\max}$, where $W_{\max}$ is the maximum absolute row sum of the matrix. Consequently, for a fixed length L, the squared norm is bounded by $\|\hat{\phi}\|_2^2 \leq \frac{\alpha_L^2}{p^2(1-p)^{2L}} W_{\max}^{2L}$.

Taking the expectation over all possible walk lengths $\ell \sim \mathrm{Geom}(p)$ and substituting the Taylor coefficients $\alpha_\ell = \frac{(-\beta/2)^\ell}{\ell!}$, we have:

$$\mathbb{E}\left[\|\hat{\phi}\|_2^2\right] \leq \sum_{\ell=0}^{\infty} p(1-p)^\ell \frac{\alpha_\ell^2}{p^2(1-p)^{2\ell}} W_{\max}^{2\ell} = \frac{1}{p} \sum_{\ell=0}^{\infty} \frac{1}{(\ell!)^2} \left(\frac{\beta W_{\max}}{2\sqrt{1-p}}\right)^{2\ell}. \tag{10}$$

We recognize this infinite series as the exact Taylor expansion of the Modified Bessel Function of the First Kind, $I_0(z) = \sum_{\ell=0}^{\infty} \frac{1}{(\ell!)^2}(z/2)^{2\ell}$, evaluated at $z = \frac{\beta W_{\max}}{\sqrt{1-p}}$. It is a well-established property that $I_0(z)$ grows asymptotically as $I_0(z) \sim \frac{e^z}{\sqrt{2\pi z}}$. Absorbing static constants into the asymptotic notation, the expected squared length is strictly bounded by exponential growth:

$$\mathbb{E}\left[\|\hat{\phi}\|_2^2\right] \leq \mathcal{O}\left(\exp\left(\frac{\beta W_{\max}}{\sqrt{1-p}}\right)\right) \implies \mathrm{Var}(\hat{Z}_v) \leq \mathcal{O}\left(\exp\left(\frac{2\beta W_{\max}}{\sqrt{1-p}}\right)\right). \tag{11}$$

To estimate the full trace over the $N \times N$ matrix, we average M independent node estimates and scale by N. Thus, the total process variance is bounded by $\frac{N^2}{M}\mathcal{O}\left(\exp\left(\frac{2\beta W_{\max}}{\sqrt{1-p}}\right)\right)$.

By Chebyshev's Inequality, to guarantee $\mathbb{P}(|\hat{\mathcal{Z}} - \mathcal{Z}| \geq \epsilon\mathcal{Z}) \leq \delta$, we require a sample size M satisfying:

$$M = \mathcal{O}\left(\frac{\mathrm{Var}(N\hat{Z}_v)}{\epsilon^2 \delta \mathcal{Z}^2}\right). \tag{12}$$

To evaluate the denominator, we require a strict lower bound on the true target $\mathcal{Z} = \sum_{i=1}^{N} e^{-\beta E_i}$, where E_i are the energy eigenvalues of H. By the Gershgorin Circle Theorem, the maximum possible eigenvalue is strictly bounded by the maximum row sum: $E_i \leq W_{\max}$. Substituting this yields a conservative minimum state contribution, giving a total partition lower bound of $\mathcal{Z} \geq N e^{-\beta W_{\max}}$.

Substituting the numerator and denominator into the Chebyshev bound yields:

$$M = \mathcal{O}\left(\frac{N^2 \exp\left(\frac{2\beta W_{\max}}{\sqrt{1-p}}\right)}{\epsilon^2 \delta \cdot N^2 \exp(-2\beta W_{\max})}\right) = \mathcal{O}\left(\frac{1}{\epsilon^2\delta} \exp\left(2\beta W_{\max}\left(1 + \frac{1}{\sqrt{1-p}}\right)\right)\right). \tag{13}$$

Crucially, the exponential spatial dimension N^2 perfectly cancels out. Finally, for a quantum spin Hamiltonian on n sites with a maximum local coupling degree d, the maximum row sum is bounded by $W_{\max} = \mathcal{O}(nd)$. Absorbing scalar multipliers into the asymptotic formulation yields the final dimensionality-free sample complexity bound:

$$M = \mathcal{O}\left(\frac{1}{\epsilon^2\delta}\exp\left(\mathcal{O}\left(\frac{\beta nd}{\sqrt{1-p}}\right)\right)\right). \tag{14}$$

□

B Detailed Algorithms

In this section, we provide the detailed algorithms for all the systems considered in this work. .

Algorithm 3 SK Model Subroutines for HWE

Objective: Execute HWE dynamically via $\mathcal{O}(n^2)$ init and $\mathcal{O}(n)$ dense updates.

function INITWALK(v) ▷ Plugs into Alg. 1, Walk A, B
  $s \leftarrow v, \quad \ell_i \leftarrow \sum_{j\neq i} J_{ij}s_j \ (\forall i), \quad E \leftarrow -\frac{1}{2}\sum_i s_i\ell_i$
  return s, ℓ, E

function TAKESTEP(s, ℓ, E) ▷ Plugs into Alg. 1, Walk A, B
  Implicit Weights: $H_{s,s\oplus e_k} \leftarrow -h_k, \quad H_{ss} \leftarrow E$
  Transition $s \to s' = s \oplus e_k$ wp $\propto h_k$, or stay wp $\propto |E|$
  if walk flipped spin k to transition **then**
    $E' \leftarrow E + 2s_k\ell_k, \quad \ell_j \leftarrow \ell_j - 2J_{jk}s_k \ (\forall j \neq k), \quad s_k \leftarrow -s_k$
  return s, ℓ, E'

B.1 Spin Glass Systems

In this section we will provide the algorithm to compute the neighbors on the fly for a dense random Ising system.

Algorithm 3 presents an $O(n)$ incremental update scheme for traversing the configuration space of the Sherrington-Kirkpatrick (SK) model. In dense spin systems with all-to-all connectivity, naive energy evaluations require $O(n^2)$ operations per transition. We circumvent this bottleneck by maintaining a vector of local fields, $l_i = \sum_{j\neq i} J_{ij}s_j$, which serves as a sufficient statistic for the local energy gradient. Upon a single spin-flip at index k, the Hamiltonian shift ΔE is computed in constant time, while the global field consistency is restored in $O(n)$ time. This linear scaling is critical for simulating long-time trajectories and high-resolution exploration of the complex, non-convex landscapes characteristic of mean-field spin glasses.

B.2 XXZ System

In this section, we provide a detailed description of our novel random walk mechanism for the XXZ system, structured as a magnetization-preserving incremental Markov chain. Unlike standard Ising-type dynamics, the XXZ Hamiltonian $\mathcal{H}$ possesses a $U(1)$ symmetry that restricts its action to specific constant-magnetization sectors Ω_m of the Hilbert space. Our algorithm explicitly respects this physical constraint by utilizing $01 \leftrightarrow 10$ swap moves as its primary transition operators. To ensure scalability to large spin systems, we implement an incremental local field maintenance scheme, which reduces the computational complexity of the diagonal energy updates to $O(d_i + d_j)$ per transition, where d denotes the vertex degree in the coupling graph $\mathcal{J}$.

We would briefly like to mention that there is an initialization cost of $O(n^2)$. In this phase we pre-compute the local fields L_i and energy E for a single configuration after which every transition is performed in time $O(d)$. Finally if J is structured (e.g., nearest-neighbor on a lattice) then d is constant (e.g., $d = 4$ for a 2D square lattice), making each step $O(1)$ independent of the size of the system.

Algorithm 4 Unbiased Trace Estimation via Dual GRF Walks

Require: Coefficients $f(k)$ for the expansion of $\sqrt{f(H)}$, halt probability $p_{\text{halt}} \in (0, 1)$, number of Hutchinson samples N_{Hutch}, number of source nodes N_{sources}, number of walk pairs per source N_{walks}, dimension $N = 2^n$

Ensure: Unbiased estimate of $\text{Tr}(f(H))$

norm $\leftarrow \frac{N}{N_{\text{sources}} \times N_{\text{walks}}}$
$E_{\text{samples}} \leftarrow [\,]$
for $s = 1, \ldots, N_{\text{Hutch}}$ **do**
  $E_{\text{hutch}} \leftarrow 0$
  for $m = 1, \ldots, N_{\text{sources}}$ **do**
    Sample start node $v \sim \text{Uniform}(0, N-1)$
    for $w = 1, \ldots, N_{\text{walks}}$ **do**
      ▷ Generate two independent estimates for the same source v
      $\text{sum}_A \leftarrow \textsc{SampleWalk}(v, s, f, p_{\text{halt}})$ ▷ See Procedure below
      $\text{sum}_B \leftarrow \textsc{SampleWalk}(v, s, f, p_{\text{halt}})$
      $E_{\text{hutch}} \leftarrow E_{\text{hutch}} + \text{norm} \times (\text{sum}_A \times \text{sum}_B)$
  Append E_{hutch} to E_{samples}
return mean(E_{samples})

procedure SAMPLEWALK(v, s, f, p_{halt})
  $x \leftarrow v, \quad W_{\text{prod}} \leftarrow 1.0, \quad P_{\text{marg}} \leftarrow 1.0, \quad k \leftarrow 0, \quad \text{acc} \leftarrow 0$
  $\sigma, D, l \leftarrow \textsc{InitDiagonalState}(x)$
  loop
    $\mathcal{N}_{\text{off}}, \mathcal{W}_{\text{off}} \leftarrow \textsc{GetNeighborsAndWeights}(x)$
    $\mathcal{N} \leftarrow [x] \cup \mathcal{N}_{\text{off}}, \quad \mathcal{W} \leftarrow [D] \cup \mathcal{W}_{\text{off}}$
    $P \leftarrow |\mathcal{W}| / \sum |\mathcal{W}|$
    ▷ Accumulate contribution weighted by Rademacher variable $z_x^{(s)}$
    $\text{acc} \leftarrow \text{acc} + z_x^{(s)} \times \frac{W_{\text{prod}}}{P_{\text{marg}}} \times f(k)$
    Draw $u \sim \text{Uniform}(0, 1)$
    if $u < p_{\text{halt}}$ **or** $\mathcal{N}$ is empty **then break** loop
    Sample index idx from distribution P and update $x_{\text{next}} \leftarrow \mathcal{N}[idx]$
    $W_{\text{prod}} \leftarrow W_{\text{prod}} \times \mathcal{W}[idx]$
    $P_{\text{marg}} \leftarrow P_{\text{marg}} \times (1 - p_{\text{halt}}) \times P[idx]$
    if $x_{\text{next}} \neq x$ **then**
      $c \leftarrow \log_2(x \oplus x_{\text{next}})$
      $\sigma, D, l \leftarrow \textsc{FlipSpinIncremental}(c, \sigma, D, l)$
    $x \leftarrow x_{\text{next}}, \quad k \leftarrow k + 1$
  end loop
  return acc
end procedure

C Experimental Details and Hyperparameters

All evaluations were executed on a single workstation equipped with an AMD Ryzen 5 8-core processor and 32 GB of RAM, utilizing strictly CPU-based computations without hardware accelerators. Our code is available at `https://github.com/arijitthegame/hypercube_walk_estimator`.

To systematically evaluate the statistical stability and variance scaling of the HWE estimator across varying system sizes ($n \in \{6, 8, 10, 12\}$), we analyzed its performance across a grid of Monte Carlo hyperparameters. Unless otherwise specified in the ablation studies, the hyperparameter sweeps were bounded within the following ranges:

- **Number of Random Walks (R):** $\{100, 200, 500, 1000\}$
- **Hutchinson Samples (S):** $\{50, 100, 200, 500\}$
- **Sampled Source Nodes (M):** $\{64, 128, 256, 512\}$

Algorithm 5 TFIM Subroutines (Incremental State Updates)

Require: System size n, transverse fields $h \in \mathbb{R}^n$, coupling adjacency list adj

function GETNEIGHBORSANDWEIGHTS(x)
    $\mathcal{N}_{\text{off}} \leftarrow []$, $\mathcal{W}_{\text{off}} \leftarrow []$
    for $i = 0, \ldots, n-1$ **do**
        Append $(x \oplus 2^i)$ to $\mathcal{N}_{\text{off}}$
        Append $-h_i$ to $\mathcal{W}_{\text{off}}$
    return $\mathcal{N}_{\text{off}}, \mathcal{W}_{\text{off}}$

function INITDIAGONALSTATE(x)
    Initialize local fields $l \leftarrow \vec{0} \in \mathbb{R}^n$
    for $i = 0, \ldots, n-1$ **do**
        $\sigma_i \leftarrow 1 - 2 \times ((x \gg i)\ \&\ 1)$
    for $i = 0, \ldots, n-1$ **do**
        for $(j, J_{ij}) \in adj[i]$ **do**
            $l_i \leftarrow l_i + J_{ij}\sigma_j$
    $D \leftarrow -0.5 \sum_{i=0}^{n-1} \sigma_i l_i$ ▷ Compute initial diagonal entry
    return σ, D, l

function FLIPSPININCREMENTAL(i, σ, D, l)
    $D \leftarrow D + 2.0 \times \sigma_i \times l_i$
    for $(j, J_{ij}) \in adj[i]$ **do**
        $l_j \leftarrow l_j - 2.0 \times J_{ij} \times \sigma_i$
    $\sigma_i \leftarrow -\sigma_i$
    return σ, D, l

To ensure robust statistical significance, the mean relative errors and their corresponding standard errors (represented as shaded regions in all figures) were aggregated over 10 independent random sampling seeds for each configuration.

Table 3: Effective Sign and Relative Variance scaling across inverse temperature β (at $n = 10$) and system size n (at $\beta = 0.4$).

	Frustrated TFIM		XXZ Model	
β ($n = 10$)	Eff. Sign	$\mathrm{Var}/\hat{\mathcal{Z}}^2$	Eff. Sign	$\mathrm{Var}/\hat{\mathcal{Z}}^2$
0.1	0.837	1.4×10^{-6}	0.839	1.6×10^{-6}
0.5	0.561	3.4×10^{-4}	0.729	1.8×10^{-4}
1.0	0.508	1.3×10^{-2}	0.542	0.02
2.0	0.620	21.5	0.048	64.6
n ($\beta = 0.4$)	Eff. Sign	$\mathrm{Var}/\hat{\mathcal{Z}}^2$	Eff. Sign	$\mathrm{Var}/\hat{\mathcal{Z}}^2$
6	0.775	1.9×10^{-4}	0.770	9.9×10^{-5}
10	0.725	7.5×10^{-6}	0.746	7.8×10^{-5}
14	0.693	8.6×10^{-6}	1.080	2.9×10^{-4}
20	0.658	1.8×10^{-5}	0.625	4.8×10^{-4}

D Limitations

The primary limitation of the HWE estimator stems from the fundamental #P-hardness of computing quantum partition functions. While our approach strictly bounds memory and per-step time complexity to $\mathcal{O}(n + d)$, the sample complexity required to control the estimator's variance can scale exponentially with the system size n in worst-case scenarios, such as near zero-temperature ($\beta \to \infty$) or in highly frustrated regimes. However, as demonstrated in our variance scaling experiment, this

Algorithm 6 On-the-Fly Navigation in Dense SK Spin System

Require: Number of spins n, couplings $J \in \mathbb{R}^{n \times n}$, field $h \in \mathbb{R}^n$, initial state $x \in \{0,1\}^n$
Ensure: Sequence of visited states and energies

Convert x to spin vector $s \in \{-1,+1\}^n$ via $s_i \leftarrow 1 - 2x_i$
Initialize local fields:

$$\ell_i \leftarrow \sum_{j \neq i} J_{ij} s_j \quad \forall i$$

Compute initial energy:

$$E \leftarrow -\frac{1}{2} \sum_i s_i \ell_i$$

for each step $t = 1, 2, \ldots$ **do**
    Select spin index $k \in \{1, \ldots, n\}$
    (Neighbor generation)
    Flip spin: $s_k \leftarrow -s_k$
    (Incremental energy update)
    $\Delta E \leftarrow 2 s_k \ell_k$
    $E \leftarrow E + \Delta E$
    (Update local fields in $O(n)$)
    for each $j \neq k$ **do**
        $\ell_j \leftarrow \ell_j - 2 J_{jk} s_k$
    $\ell_k \leftarrow -\ell_k$
    Record (x, E)

statistical noise can often be effectively suppressed in practice by scaling the trivially parallelizable Monte Carlo hyperparameters.

Finally since HWE relies on importance weighting to traverse edges with negative weights, highly frustrated systems or near-zero temperatures ($\beta \to \infty$) will induce paths with large alternating signs. Consequently, the variance of the estimator $\hat{\mathcal{Z}}$ will grow. Achieving an ϵ-accurate estimate in these worst-case scenarios may require the number of walks R and Hutchinson samples S to scale exponentially with the system size n. However, as demonstrated empirically in Section 4, this variance can often be suppressed via trivial parallelization for practical system sizes.

E Broader Impacts

This work introduces a foundational algorithmic improvement for the simulation of quantum spin Hamiltonians. By circumventing traditional explicit memory bottlenecks, it offers positive long-term impacts for computational research in quantum many-body physics, condensed matter, and materials science, potentially accelerating the in-silico discovery of novel materials. Given the strictly theoretical and mathematical nature of this algorithmic framework, we do not foresee any direct paths to negative societal impacts, malicious misuse, or privacy risks.

Algorithm 7 Incremental Unbiased Markov Chain for XXZ Trace Estimation

Require: State $x \in \{0,1\}^n$, Coupling $\mathcal{J}$, Anisotropy Δ, Halting prob p_{halt}
Ensure: Walk sequence $\mathcal{W} = \{(x_k, \mathbf{P}_k, \mathbf{W}_k)\}_{k=0}^{L}$

Initialization:
$\sigma_i \leftarrow 1 - 2x_i$ ▷ Map bits to physical spins $\sigma \in \{-1, 1\}^n$
$L_i \leftarrow \sum_{j \neq i} \mathcal{J}_{ij}\sigma_j$ ▷ Initial local field vector $L \in \mathbb{R}^n$
$E \leftarrow \Delta \sum_{i<j} \mathcal{J}_{ij}\sigma_i\sigma_j$ ▷ Initial diagonal energy in $O(n^2)$
$\mathcal{W} \leftarrow \emptyset$
while True **do**
    1. Generate $U(1)$-Preserving Transitions:
    $\mathcal{N} \leftarrow \emptyset, \mathcal{W}_{off} \leftarrow \emptyset$
    for all pairs (i, j) such that $x_i = 0, x_j = 1$ **do**
        ▷ **Conservation Law:** Only swap opposite spins to stay in sector Ω_m
        $x' \leftarrow x \oplus (2^i + 2^j)$ ▷ Generate bitwise swap neighbor
        $\mathcal{N} \leftarrow \mathcal{N} \cup \{x'\}, \quad \mathcal{W}_{off} \leftarrow \mathcal{W}_{off} \cup \{2\mathcal{J}_{ij}\}$
    2. Markov Transition Kernel Calculation:
    $\mathcal{X}_{cand} \leftarrow \{x\} \cup \mathcal{N}, \quad \mathbf{W} \leftarrow [E] \cup \mathcal{W}_{off}$
    $\mathbf{P} \leftarrow |\mathbf{W}| / \sum |\mathbf{W}|$ ▷ Construct unbiased transition probabilities
    $\mathcal{W} \leftarrow \mathcal{W} \cup \{(x, \mathcal{X}_{cand}, \mathbf{W}, \mathbf{P})\}$
    3. Termination Check:
    if $\text{Uniform}(0,1) < p_{\text{halt}}$ **then**
        break
    4. State Update and Incremental Local Field Maintenance:
    $x_{next} \leftarrow$ Sample from $\mathcal{X}_{cand}$ via $\mathbf{P}$
    if $x_{next} \neq x$ **then**
        $\mathcal{F} \leftarrow \{i, j\}$ ▷ Identify the two flipped indices
        for $b \in \mathcal{F}$ **do**
            ▷ **Local Field Maintenance:** Update energy in $O(d)$ time
            $E \leftarrow E - (\Delta \cdot \sigma_b \cdot L_b)$ ▷ Subtract old interaction energy
            $\sigma_b \leftarrow -\sigma_b$ ▷ Flip spin at site b
            for all $k \in \text{Adj}(b)$ **do**
                $L_k \leftarrow L_k + 2\sigma_b\mathcal{J}_{bk}$ ▷ Propagate change to neighbor fields
            $E \leftarrow E + (\Delta \cdot \sigma_b \cdot L_b)$ ▷ Add new interaction energy
        $x \leftarrow x_{next}$
end while
return $\mathcal{W}$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's scope, including the $\mathcal{O}(n + d)$ complexity and the introduction of the HWE estimator.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss the limitations of our estimator regarding sample complexity and variance scaling relative to the #P-hard nature of the problem in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Theorem 3.2 provides a complete proof of the unbiasedness of the GRF-Hutchinson estimator, and structural algorithms are detailed in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully disclose hyperparameters, algorithmic details, and system configurations required to reproduce the experiments in Section 4 and the Appendix. Moreover we provide detailed algorithms for all the systems in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code is available at `https://github.com/arijitthegame/hypercube_walk_estimator`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify our experimental hyperparameters, including the number of walks, Hutchinson samples, and source nodes in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report mean relative error alongside shaded regions representing the standard error calculated across multiple random sampling runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were run on a Ryzen 5 16-core processor with 32 GB RAM.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research complies with the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a discussion regarding the broader impacts of our work in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work investigates theoretical quantum simulation algorithms and does not release high-risk models or scraped datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: We did not use proprietary third-party assets; experimental baselines were implemented using standard open-source mathematical libraries.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or pre-trained models.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: No LLMs were used for the core methodological or algorithmic development of this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.