

Rapid ground state energy estimation with a Sparse Pauli Dynamics-enabled Variational Double Bracket Flow

Chinmay Shrikhande,¹ Arnab Bachhar,¹ Aarón Rodríguez Jiménez,¹ and Nicholas J. Mayhall^{1,*}

¹*Department of Chemistry, Indiana University, Bloomington, IN 47401, USA*

Ground state energy estimation for strongly correlated quantum systems remains a central challenge in computational physics and chemistry. While tensor network methods like DMRG provide efficient solutions for one-dimensional systems, higher-dimensional problems remain difficult. Here we present a variational double bracket flow (vDBF) algorithm that leverages Sparse Pauli Dynamics, a technique originally developed for classical simulation of quantum circuits, to efficiently approximate ground state energies. By combining greedy operator selection with coefficient truncation and energy-variance extrapolation, the method achieves less than 1% error relative to DMRG benchmarks for both Heisenberg and Hubbard models in one and two dimensions. For a 10×10 Heisenberg lattice (100 qubits), vDBF obtains accurate results in approximately 10 minutes on a single CPU thread, compared to over 50 hours on 64 threads for DMRG. For an 8×8 Hubbard model (128 qubits), the speedup is even more pronounced. These results demonstrate that classical simulation techniques developed in the context of quantum advantage benchmarking can provide practical tools for many-body physics.

I. INTRODUCTION

The simulation of quantum many-body systems provides an indispensable tool for studying molecules and materials, often revealing features that are difficult or impossible to access experimentally. Improving our ability to perform such simulations is therefore one of the most important computational challenges in physics and chemistry, with direct implications for problems ranging from high-temperature superconductivity to heterogeneous catalysis. Unfortunately, strongly correlated systems remain extremely challenging, and no general-purpose classical method exists that can provide both reliable and efficient results for arbitrary systems.

Most traditional techniques for approximating low-energy quantum systems fall into one of three categories: perturbative methods (MPn [1], CCn [2–5], GW [6–8], etc.), stochastic methods (DMC [9], FCI-QMC [10, 11], AF-QMC [12–14], etc.), or tensor network states (DMRG [15–21], PEPS [22–27], MERA [28, 29], etc.). Each framework excels for certain classes of systems but has fundamental limitations. Perturbative methods are effective when mean-field treatments are qualitatively correct but break down when many-body interactions become strong. Stochastic methods can provide benchmark-quality results for ground state energies but are limited when the sign problem cannot be controlled [30]. Tensor network states provide ideal compactness for low-dimensional systems but become less efficient in higher dimensions. Despite the breadth of problems treatable by these and other methods, systems that combine strong correlation with high dimensionality remain challenging for classical simulation.

In light of rapid advances in quantum information science, quantum computation has become an appealing ap-

proach for such classically challenging problems. Quantum hardware continues to improve [31–33], with growing qubit counts and decreasing error rates, while algorithmic developments have reduced the quantum resources required for accurate simulations. However, current quantum hardware and algorithms are not yet capable of outperforming classical methods for many-body simulation, and the realization of an exponential speedup remains uncertain [34].

While quantum simulation has not yet demonstrated a definitive advantage over classical methods, its development has driven significant advances in classical simulation techniques. The 2023 IBM Eagle experiment [32], which claimed evidence for quantum utility in simulating kicked Ising dynamics, sparked rapid development of classical methods that ultimately matched or exceeded the quantum results [35–37]. Central to these classical responses were algorithms for approximating Heisenberg evolution of Pauli operators (Sparse Pauli Dynamics, Pauli Propagation, and related methods) which proved remarkably effective at simulating deep quantum circuits [35, 36, 38–50]. These tools were developed to benchmark quantum hardware, but their potential for classical ground state energy calculations remains largely unexplored. In this work, we combine Sparse Pauli Dynamics with a variational double bracket flow (vDBF) approach for ground state energy estimation. For challenging two-dimensional systems, a 10×10 Heisenberg lattice and an 8×8 Hubbard model, vDBF achieves sub-1% accuracy while requiring orders of magnitude less computational time than DMRG.

II. THEORY

To simulate the results of a quantum computation, one generally aims to compute the expectation values of observables for states that have been evolved by a sequence

* nmayhall@iu.edu

of unitaries,

$$\langle O \rangle = \langle 0 | U_1^\dagger U_2^\dagger \cdots U_n^\dagger O U_n \cdots U_2 U_1 | 0 \rangle. \quad (1)$$

While this can be expressed equivalently in either the Schrödinger or Heisenberg picture, the choice to evolve the state, $\langle \psi_n | O | \psi_n \rangle$, or the operator, $\langle 0 | O_n | 0 \rangle$, or both[36] presents distinct opportunities for approximations.

Due, in large part, to the fact that the wavefunction is generally a lower-rank object, the Schrödinger picture is typically preferable. However, considerations of sparsity can often lean in favor of the Heisenberg picture. Physically relevant Hamiltonians are generally sparse, while ground states are generally not.[51] Furthermore, when the state is no longer pure (i.e., noisy circuits or finite temperature), a density operator is needed, further reducing the computational distinction between the Heisenberg and Schrödinger pictures.

Choosing Pauli strings, P_i , (defined as the positive Hermitian tensor products of Pauli operators) as an operator basis, we can identify several computational advantages for working in the Heisenberg picture arising from their extremely simple and well-characterized mathematical properties.[52–55] Paulis can be conveniently indexed in binary form, and operator multiplication can be efficiently evaluated through simple binary operations on integers. Because any two Pauli strings either commute or anticommute, these can also be efficiently obtained through binary operations. Due to the involution property, Pauli strings have closed form exponentiation, making evolution of a state,

$$U_j(\theta) |\psi\rangle = e^{-i\frac{\theta}{2}P_j} |\psi\rangle = \cos\left(\frac{\theta}{2}\right) |\psi\rangle - i \sin\left(\frac{\theta}{2}\right) P_j |\psi\rangle \quad (2)$$

or Pauli string,

$$U_j(\theta)^\dagger P_k U_j(\theta) = \begin{cases} \cos(\theta) P_k + i \sin(\theta) P_j P_k & \text{if } [P_k, P_j] \neq 0 \\ P_k & \text{if } [P_k, P_j] = 0 \end{cases} \quad (3)$$

exceptionally convenient. More general considerations of exact exponentiation have recently been explored by Izmaylov and coworkers [56] and Evangelista and Magoulas [57]. Furthermore, the Heisenberg picture Pauli evolution naturally exposes simplifications from Clifford or low-magic unitaries. While a $\pi/2$ rotation about any Pauli string has the potential to create a maximally entangled state in the Schrödinger picture, this angle corresponds to a Clifford gate, which, by definition, maps Pauli strings to Pauli strings, leaving the coefficient structure (and thus sparsity pattern) of any Heisenberg operator unchanged. For example, compare the action of a $\pi/2$ rotation on either a state with an operator:

$$U_i\left(\frac{\pi}{2}\right) |\psi\rangle = \frac{1}{\sqrt{2}} (|\psi\rangle - i |P_i \psi\rangle) \quad (4)$$

$$U_i\left(\frac{\pi}{2}\right)^\dagger P_j U_i\left(\frac{\pi}{2}\right) = P_k, \quad (5)$$

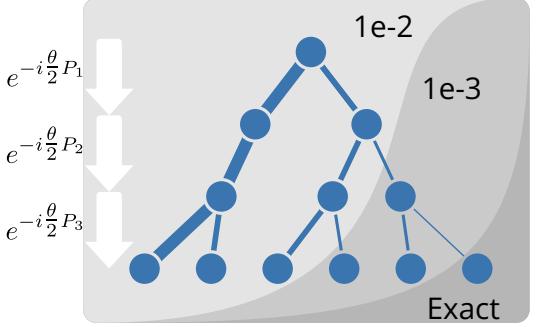


FIG. 1: Schematic depiction of the evolution of a single Pauli string (top) following 3 sequential small-angle Pauli rotations. Background colors indicate the additional set of Paulis that are included under tighter levels of coefficient truncation.

assuming P_i and P_j don't commute. In the Schrödinger picture, this Clifford evolution of a state increases in complexity generally, whereas the Pauli string, P_j , is just converted to a different Pauli, P_k .

A. Pauli Evolution Truncation

Simulating the sequential rotation of an operator by a large number of Pauli strings, Eq. 1, results in an exponentially increasing number of Paulis. However, Eq. 3 illustrates that this Pauli sum will potentially have a considerable amount of structure. Each rotation splits each non-commuting Pauli in the Pauli sum into two branches, a cosine and a sine branch, forming a binary tree. If each rotation angle is small, then the cos branches dominate, with the path weights diminishing exponentially in the number of sin branches. In contrast, for near-Clifford rotations ($\theta \approx \pi/2$), the sin branch dominates. The schematic in Fig. 1 illustrates the operator spreading that occurs during sequential evolution. Pauli evolution, therefore, creates a binary tree, such that the evaluation of the final expectation value requires one to sum the expectation values of all the leaves, weighted by the weight of the branch. While this is clearly exponential to do exactly, the fact that many of these paths will have extremely small weights leads to a number of truncation options.

One could take a depth-first search approach to this, evaluating the sum path-by-path and aborting a given path once the current weight drops below a threshold. This requires essentially zero memory and is embarrassingly parallelizable; however, it performs rather poorly due to the need to go to very tight thresholds for convergence. One reason for this is that each path can visit the same Pauli, resulting in an opportunity for interference between the paths. Instead, we use a breadth-first search approach where the full operator at each time step is stored in memory, providing the opportunity to account for interference between Pauli paths. Keeping

track of the full operator stabilizes truncation and facilitates the computation of properties such as 4-point correlation functions. In this paper, we perform truncations at each time step, deleting all Paulis whose coefficient absolute value is smaller than a given threshold ϵ . This approach is not new and has been used in several implementations, including Refs. [49] and [35].

The approximation of Pauli Heisenberg dynamics has been described with a variety of names, Pauli Propagation,[38, 49], Pauli Backpropagation,[58], Pauli Path simulation,[50], and Sparse Pauli Dynamics (SPD).[35], each with different, but related goals. Here, we refer to our usage of approximate Pauli evolution as SPD, since we are currently only approximating the evolution by truncating small coefficient Paulis, leaving the analysis of Pauli weight truncation for later work.

B. Double Bracket Flows

Having access to an efficient algorithm for approximating unitary transformations has the potential to be useful for a variety of applications. The utility for quantum circuit simulation is clear and well-documented. In this work, we are interested in applying SPD to the problem of ground state energy estimation.

In order to define a unitary transformation that evolves a state into the ground state, we start by adopting a double bracket flow formalism. Double bracket flows are differential equations whose right-hand sides involve nested commutators, and which, when integrated, drive the system toward specific targets. This approach was originally introduced as the Similarity Renormalization Group (SRG) in 1993 by G  azek and Wilson,[59, 60] and was independently developed as a flow equation by Wegner in 1994.[61]

The basic idea is to define a Hamiltonian transformation along a continuous time-like parameter, s , that moves in a direction that minimizes some cost function. The “canonical commutator” [59, 61] is defined using the commutator of the Hamiltonian with the projection onto its own diagonal, H_D :

$$\frac{d}{ds}H(s) = [G(s), H(s)] = [[H_D(s), H(s)], H(s)]. \quad (6)$$

Since $G(s)$ is an anti-Hermitian flow generator operator, this is analogous to a unitary evolution of H by a time-dependent drive $G(s)$, a process that preserves the spectrum of $H(s)$ at all times, and is thus a similarity transformation. This choice of $G(s)$ generates rotations that minimize the off-diagonal components, and goes to zero when the Hamiltonian commutes with its own diagonal. This form of $G(s)$ has been widely used, finding applications in physics,[62] chemistry,[63–65] and quantum algorithms.[48, 66]

1. State-specific flow generator

While the “canonical commutator” would certainly suffice for obtaining our target eigenstate because it achieves full diagonalization, it may be unnecessarily powerful, and we thus choose a flow generator that targets a single reference state instead, $G(s) = [H(s), \rho]$. This generates evolution anytime the Hamiltonian does not have ρ as a stationary state. We choose ρ to be the projector onto a reference computational basis state, which we will take, without loss of generality,[67] to be the zero state, $\rho = |0\rangle\langle 0|$. Since the reference state expectation value of the right-hand side is just the energy variance, which is non-negative, the energy is guaranteed to decrease until $|0\rangle$ becomes an eigenstate:

$$\frac{d}{ds}E(s) = -2(\langle H^2 \rangle - \langle H \rangle^2) \quad (7)$$

While this natural choice converges well, the projector expanded in the Pauli basis is highly non-sparse, requiring 2^N diagonal Pauli strings,

$$|0\rangle\langle 0| \propto I + \sum_i Z_i + \sum_{i < j} Z_i Z_j + \dots \quad (8)$$

To avoid this exponential cost, we choose to truncate this projector, where we approximate $|0\rangle\langle 0|$ by keeping only Z strings up to a certain Pauli weight. For all results presented here, we keep only the 1-body terms. While this might seem like a dramatic approximation, evolving by only the 1-body terms will still work, as this generator will only go to zero once $H(s)$ becomes fully diagonal. However, with this approximation, the integration of this equation takes longer to converge, as demonstrated in Fig. 6 of Appendix A. Using this 1-body form, our double bracket flow generator becomes:

$$G(s) = \sum_i [H(s), Z_i], \quad (9)$$

leading to the following double bracket flow:

$$\frac{d}{ds} H(s) = \sum_i [[H(s), Z_i], H(s)]. \quad (10)$$

Integrating Eq. 10 for a small time, δs , we obtain the following recursive transformation:

$$H(s + \delta s) = e^{G(s)\delta s} H(s) e^{-G(s)\delta s} \quad (11)$$

Since our aim will be to approximate this process using SPD, we will want to first expand $G(s)$ in the Pauli basis:

$$G(s) = \sum_i g_i P_i, \quad (12)$$

where g_i depend on s and are purely complex. Since δs is small, we can expand this solution in a product formula:

$$H(s + \delta s) = e^{g_n P_n \delta s} \dots e^{g_1 P_1 \delta s} H(s) e^{-g_1 P_1 \delta s} \dots e^{-g_n P_n \delta s} \quad (13)$$

Using the equation above, we *could* proceed by using SPD to sequentially evolve $H(s)$ by the series of Paulis in $G(s)$, each time truncating $H(s)$ by deleting Paulis with magnitudes smaller than ϵ . However, keeping δs small (a requirement to explicitly integrate Eq. 10) would require too many rotations for efficient convergence. In the next section, we propose a modification of this strategy that accelerates this convergence.

C. Sparse Pauli Dynamics - Variational DBF (vDBF)

At this point, we depart from the direct integration of the double bracket flow defined in Eq. 10. Rather than choosing the time steps to be “small”, which would inevitably require many Pauli rotations to converge, we will instead choose an *optimal* angle, θ_i^* , for each Pauli, P_i , in the sequence, chosen to minimize the energy of the reference state, $|\psi\rangle$, by analytically minimizing the cost function

$$\begin{aligned} F(\theta_i) &= \langle\psi| e^{iP_i\theta_i} H e^{-iP_i\theta_i} |\psi\rangle \\ &= \cos^2(\theta_i) \langle\psi| H |\psi\rangle + \sin^2(\theta_i) \langle\psi| P_i H P_i |\psi\rangle \\ &\quad - 2i \cos(\theta_i) \sin(\theta_i) \langle\psi| H P_i |\psi\rangle \end{aligned} \quad (14)$$

This will ensure that each Pauli rotation minimizes the energy by the maximum amount possible. While this leads to a trajectory that deviates from the specific double bracket flow defined in 10, we ultimately only care about the final state. Since we are no longer faithful to the flow of Eq. 10, the s parameter is no longer meaningful, so we track the progress of this sequential transformation with an index, indicating the number of iterations: $H(s) \rightarrow H^{(i)}$.

1. Greedy selection of generators

Depending on the number of rotations performed during the vDBF, $H^{(i)}$ can accumulate many terms, resulting in a flow generator for the current iteration, $G^{(j)} = \sum_i [H^{(j)}, Z_i]$, that is even larger. However, not all Paulis in $G^{(j)}$ are equally important. Ideally, we would only rotate by the most “variationally helpful” generators within each iteration to avoid unnecessary computation. To do this, we sort the operators in $G^{(i)}$ by the derivative of the energy with respect to rotation about each operator:

$$\frac{d}{d\theta_i} \langle\psi| e^{i\theta_i g_i P_i} H^{(n)} e^{-i\theta_i g_i P_i} |\psi\rangle = g_i \langle\psi| [P_i, H^{(n)}] |\psi\rangle \quad (15)$$

and only rotate by the largest `n_rots` (user-specified convergence parameter) angles per iteration. Setting `n_rots`=1 would therefore result in a single rotation by the operator with the largest gradient in the $G^{(i)}$. This

would be a dynamic analog to how ADAPT-VQE [68, 69] chooses operators out of a fixed operator pool. However, since evaluating the flow generator is not free, we find that it’s generally more efficient to perform several rotations for each iteration, and so choosing `n_rots` to be tens to hundreds (e.g., 50-500) seems to work well. We note that this algorithm for designing a double bracket flow, which uses optimal steps to converge to the ground state, can be viewed as a classical version of the related quantum algorithm, Double Bracket Diagonal Operator Iterations.[48]

2. Operator Truncation

In order to tame the exponential growth of the Hamiltonian, we follow the coefficient truncation schemes used in both SPD[35, 36, 43] and PauliPropagation,[45–47, 49] by which an operator expressed as a linear combination of Paulis, $O = \sum_i c_i P_i$, is modified by deleting all Paulis, P_i , where $|c_i| < \epsilon$. We refer to this truncation as CLIP in Line 19 of the Algorithm 1. While Pauli weight[44, 49] (or Majorana weight for fermionic Hamiltonians)[47] can also be used to further decrease computational cost, we delay an in-depth analysis of this for future work (initial calculations seem to suggest it can be helpful). [70]

3. Estimating “lost” energy

While the operator truncation occurring in the CLIP function is obviously necessary to prevent the size of the operator from becoming intractable, it does result in some amount of error, which grows with the number of rotations. One simple way to reduce the impact on the final result is to calculate the impact that truncation has on the energy with respect to our reference state. By computing the expectation value before and after CLIP, we obtain the energy lost to truncation, which we can sum over all the iterations to obtain an energy correction for the final operator. Since the computation of expectation values is rather cheap, this is essentially a free way to significantly improve the resulting energy, and we refer to this as the “corrected energy”.

4. Extrapolation

Algorithm 1 lists the steps required to converge a vDBF calculation for a given, user-specified accuracy control, ϵ . By decreasing the truncation threshold from fast, loose calculations (e.g., $\epsilon \approx 1e-2$) to slower, tighter calculations (e.g., $\epsilon \approx 1e-3$ to $1e-5$), one can systematically increase the accuracy of the calculations. While in principle, one could simply tighten these parameters until the desired accuracy is reached, this is limited by two practical considerations, the inevitable exponential scaling of the Hamiltonian, and the slow convergence that

Algorithm 1: vDBF

Input: $H, \psi, \epsilon, n_rots, max_iter, conv_thresh$
Output: $H^{(i)}, generators, angles$

```

1 generators ← [];
2 angles ← [];
3  $H^{(1)} \leftarrow H;$ 
4 for  $i \leftarrow 1$  to  $max\_iter$  do
    // Compute generator
    5  $G^{(i)} \leftarrow [H^{(i)}, \sum_i Z_i];$ 
    // Sort G by gradient
    6 gradients ← [];
    7 paulis ← [];
    8 for  $g_i, P_i \in G^{(i)}$  do
        | Append  $g_i \times \langle \psi | [P_i, H^{(i)}] | \psi \rangle$  to gradients;
        | Append  $P_i$  to paulis;
    9 perm ← sortperm(gradients);
    10 paulis ← paulis[perm];
    // Check for convergence
    11 if  $norm(gradients) < conv\_thresh$  then
        | break;
    // Rotate by optimal angles
    15 for  $j \leftarrow 1$  to  $n\_rots$  do
        |  $P_j \leftarrow paulis[j];$ 
        |  $\theta_j^* \leftarrow COMPUTE\_OPTIMAL\_ANGLE(H^{(i)}, \psi, P_j);$ 
        |  $H^{(i)} \leftarrow EVOLVE(H^{(i)}, P_j, \theta_j^*);$ 
        |  $H^{(i)} \leftarrow CLIP(H^{(i)}, \epsilon);$ 
        | Append  $\theta_j^*$  to angles;
        | Append  $P_j$  to generators;
    21  $H^{(i+1)} \leftarrow H^{(i)}$ 
22
23 return  $H^{(i)}, generators, angles$ 

```

occurs when optimizing a single parameter at a time. This leads to two distinct errors: (i) *truncation errors* which arise from discarding Paulis with small coefficients and (ii) *convergence errors* which arise from aborting the optimization before the energy is fully minimized (or variance has reached zero). In order to estimate the energy obtained in the exact limit (i.e., setting $\epsilon = 0$ and completely minimizing the energy), one can perform an energy extrapolation in one of two ways:

a. *Corrected Energy vs Discarded Weight* Using this strategy, one can perform a series of calculations with a range of ϵ values, to obtain an approximately linear relationship between the energy and the “discarded weight”, which is computed as

$$DW^{(i)} = ||CLIP(H^{(i)}, \epsilon)||_F^2 - ||H^{(0)}||_F^2, \quad (16)$$

and provides a direct measure of how much the operator has been truncated (i.e., how non-unitary was the transformation). This is analogous to the manner in which DMRG calculations are often extrapolated. However, this would require each calculation to be run till convergence. As we will see below, this is often difficult to do in practice, and one would either need to use

loose thresholds for which convergence is easier, or employ a two-layer extrapolation, where convergence errors are corrected first, followed by zero discarded weight extrapolation.

b. *Energy vs Variance* Alternatively, we can perform an extrapolation that simultaneously accounts for both truncation and convergence errors. At each iteration of vDBF, we can compute the energy variance $\langle H^2 \rangle - \langle H \rangle^2$ with respect to the reference state, providing a rigorous measure of the distance from the nearest eigenstate. Once the iterations are close enough to convergence such that the relationship between energy (or “corrected energy”) and variance becomes linear, one can simply extrapolate from the data obtained from a single vDBF calculation to obtain an estimate for the energy at convergence. However, this extrapolated energy relates to the energy of the converged, yet truncated, Hamiltonian. Alternatively, we could also compute the accumulated error in the variance by calculating the variance before and after each CLIP, just as we described for the accumulated energy error in Sec. II C 3. By plotting the “corrected energy” vs the “corrected variance” and extrapolating, one obtains an estimate of the energy, not only at the convergence limit, but also at the *zero truncation* limit. This is a particularly efficient way to obtain final estimations of the energy from only a single calculation. Unlike the computation of the accumulated energy error, the accumulated variance error does come at a computational cost, which often ends up doubling the total cost of a vDBF calculation, albeit providing a much more accurate result.

D. Comparison to existing methods

The vDBF method is related to several other simulation techniques, both quantum and classical.

a. *ADAPT-VQE*: In ADAPT-VQE,[68, 69] one performs a variational minimization of an ansatz that has the form of a sequence of unitary transformations. The specific sequence of generators is defined iteratively such that at each iteration, the ground state energy gradient is used as an importance measure for determining which operator from a static, user defined operator pool should be added to the ansatz. After adding each new generator, all the operators are allowed to variationally relax. Our current vDBF algorithm can be viewed as a classical realization of a modified ADAPT-VQE, where only the last operator added is minimized (avoiding the challenging non-linear optimization), and the operators are chosen from a dynamic pool, $[H(s), \sum_i Z_i]$, which grows and evolves with the algorithm.

b. *iterative Qubit Coupled Cluster (iQCC)*: vDBF can be viewed as a classical version of iQCC[71, 72] where the entanglers are chosen based on the flow generator in Eq. 9 instead of the direct interaction space, and the evolution is approximated using SPD. A recent preprint has also demonstrated a multistate variant, which showed

promising results on small systems.[73]

c. DB-DOI: In the recently proposed quantum algorithm Double Bracket-Diagonal Operator Iteration (DB-DOI), the Hamiltonian is diagonalized through a double bracket flow on a quantum computer,[48] where each sequential rotation can be variationally determined to minimize the off-diagonal of the Hamiltonian. The current vDBF method can be viewed as a classical version of the DB-DOI approach, where the time steps are chosen to minimize the expectation values of the reference state, and the time evolution is approximated using the SPD technique.

III. NUMERICAL SIMULATIONS

In order to assess the ability of vDBF to approximate ground state energies, we have selected two canonical quantum many-body problems as examples: an antiferromagnetic Heisenberg spin lattice and the Fermi Hubbard model. In both systems, we demonstrate the vDBF method's ability to provide fast estimates of ground-state energies.

a. Extrapolation procedure To obtain stable extrapolations to the exact limit, we leverage the fact that when the trajectory is near convergence, we expect a linear relationship between the energy and the energy variance.[74–77] However, further from convergence, the deviation from linearity might be non-negligible, and a simple linear regression might be less reliable despite having a seemingly respectable R^2 value. In order to obtain a more conservative estimate of errors in our extrapolations, we take the extrapolated result to be the average of a linear (b_1) and a quadratic (b_2) fit, with the uncertainties reported as half the difference. While we find that this extrapolation works surprisingly well, the fact that each calculation often generates thousands of data points means that one must decide how many data points to include in the extrapolation.

In the calculations below, we start with the final step of a given optimization trajectory (which occurs due to either convergence or reaching the maximum number of iterations allowed), and then include all the previous points up to a cutoff which is chosen to minimize the sum of the uncertainty $((b_1 - b_2)/2)$ and the R^2 value of the linear fit. This balances precision with goodness of fit. Using this “optimal” cutoff point in our data, our zero-variance extrapolations are reported as the average of the linear and quadratic fits, with the difference as the uncertainty: $E_{(v=0)}^\infty = \frac{b_1+b_2}{2} \pm \frac{b_1-b_2}{2}$.

All vDBF calculations were performed with our own software, `DBF.jl` and `PauliOperators.jl`[78] which use standard symplectic Pauli representations for efficient binary manipulations. All DMRG calculations were carried out using the `ITensors.jl` package.[79]

A. Heisenberg Hamiltonian

The isotropic Heisenberg Hamiltonian is a standard model not only for describing magnetic interactions in materials but also for evaluating computational methods for strongly correlated quantum systems. While the one-dimensional lattice admits an analytic solution via the Bethe ansatz, and DMRG provides an optimally efficient numerical treatment, the simulation in higher dimensions has no analytic solution, and precludes efficient simulation with a 1D MPS. While belief propagation algorithms for general tensor network state contractions have resulted in significant improvements recently,[37, 80–82] the description of higher dimensional strongly correlated systems is still challenging.

We start by computing the ground state energy of the antiferromagnetic Heisenberg spin Hamiltonian:

$$H = -2J \sum_{\langle ij \rangle} \vec{S}_i \cdot \vec{S}_j \quad (17)$$

on a series of lattices: (a) a large one-dimensional 1×100 lattice, (b) a small two-dimensional 4×4 lattice, and (c) a large two-dimensional 10×10 square lattice. For the DMRG calculations, we used the `ITensors.jl` software,[79] and ordered the sites using the Fiedler vector (taking the coupling matrix as our adjacency matrix), for which we observed faster convergence than with a snake pattern.

For the vDBF results, we choose the Néel state for our reference state $|0101\dots\rangle$. Because the form of the projector, ρ , and the resulting approximate flow generator (Eq. 9) assume the zero state, we simply transform our initial Hamiltonian by a sequence of X gates so that we can work directly with the zero state:

$$E^{(0)} = \langle 0101\dots | H^{(0)} | 0101\dots \rangle \quad (18)$$

$$= \langle 0 | \dots X_4 X_2 H^{(0)} X_2 X_4 \dots | 0 \rangle \quad (19)$$

Since each X_i gate is a Clifford, the structure of $H^{(0)}$ is left unchanged. This feature will allow us to consider entangled reference states in future work.

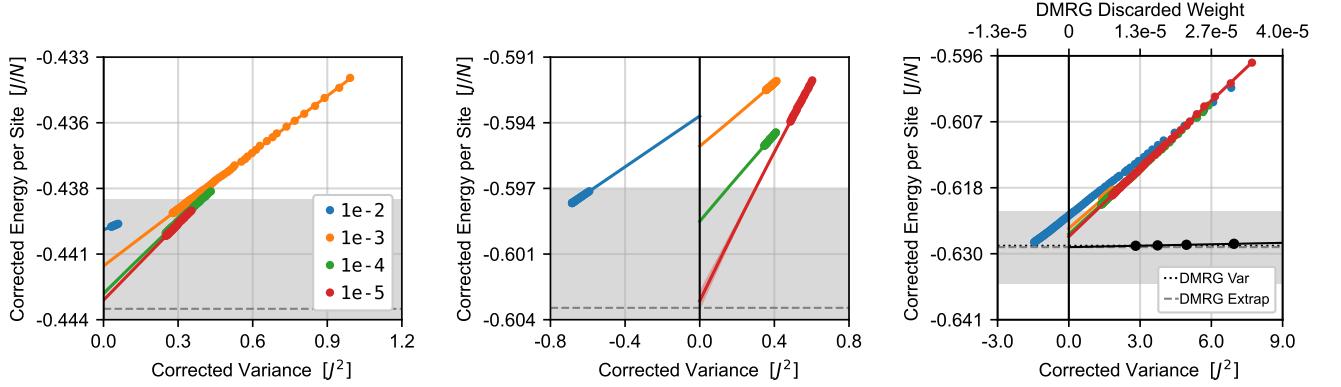
The results for each of the three Heisenberg lattices are presented in Table I, and complementary convergence plots are provided in Fig. 2.

1. 1×100 Heisenberg

Looking first at the one-dimensional 1×100 Heisenberg lattice, we find that all 4 thresholds achieve less than 1% error, with $\epsilon = 1e-2$ doing so in just a few seconds on a single thread. We observe systematic accuracy improvement upon decreasing ϵ ; each tighter calculation in the sequence decreases the deviations from DMRG, up to the $\epsilon = 1e-5$ result, which has less than 0.1% error. While both the energy errors and discarded weight decrease as ϵ is decreased, the cost also increases. It appears from

Heisenberg Lattice		DMRG	vDBF				
			$\epsilon :$	1e-2	1e-3	1e-4	1e-5
1x100	100	Energy/Site: -0.443230	Energy/Site: -0.440021(3)	-0.441488(6)	-0.4426(1)	-0.44289(2)	
			Error:	0.72%	0.39%	0.14%	0.08%
			Variance:	0.0115	0.0924	0.2201	0.2472
			Discarded W.:	0.720	0.063	0.006	0.0005
		$\chi:$ 800	Iterations:	100	100	100	100
		Time(min): 3.9	len(H):	2641	33462	314410	2770908
		Threads: 64	Time(min):	0.5	4.3	22.7	172.8
			Threads:	1	1	1	1
6x6	36	Energy/Site: -0.603522	Energy/Site: -0.59378(2)	-0.59532(9)	-0.59913(9)	-0.6032(5)	
			Error:	1.61%	1.36%	0.73%	0.05%
			Variance:	0.002	0.093	0.230	0.467
			Discarded W.:	1.013	0.156	0.036	0.003
		$\chi:$ 1200	Iterations:	100	100	100	48
		Time(min): 77.1	len(H):	2287	61898	1348604	16128678
		Threads: 64	Time(min):	0.2	6.1	114.4	579.03
			Threads:	1	1	1	1
10x10	100	Var E/Site: -0.628455 Extrap E/Site: -0.628693	Energy/Site: -0.623210(2)	-0.625448(2)	-0.62645(2)	-0.62687(3)	
			Error:	0.83%	0.45%	0.32%	0.25%
			Variance:	0.721	0.753	1.193	1.7913
			Discarded W.:	2.807	0.333	0.053	0.007
		$\chi:$ 3500	Iterations:	100	100	100	56
		Time(min): 3213.9	len(H):	8565	181011	2784666	28721856
		Threads: 64	Time(min):	0.7	10.3	184.6	845.6
			Threads:	1	1	1	1

TABLE I: Energy/Site values for Heisenberg models for DMRG and vDBF. vDBF calculations are each run with the threshold, ϵ , defined in the top row, and `max_iter`=100 and `n_rots`=100. Uncertainties in the vDBF energies, shown in parentheses, are given by half the difference between linear and quadratic extrapolation, as described in Sec. III 0 a. Discarded W. is the discarded weight from Sec. II C 4. 1x100 lattice has periodic boundary conditions. 6x6 and 10x10 have open boundary conditions.



(a) Energy vs Variance extrapolation: 1x100 (b) Energy vs Variance extrapolation: 6x6 (c) Energy vs Variance extrapolation: 10x10

FIG. 2: Energy vs Variance extrapolations for Heisenberg lattices using a series of vDBF thresholds: $\epsilon = 1\text{e-}2$ (blue), $1\text{e-}3$ (orange), $1\text{e-}4$ (green), $1\text{e-}5$ (red). (a) 1x100 lattice, (b) 6x6 lattice, and (c) 10x10 lattice. Extrapolations shown as solid lines. Uncertainties (difference between linear and quadratic fits) shown as shaded regions. DMRG best estimates (dashed grey line). 1% error region w.r.t. DMRG best estimate (grey shaded area). For (c), DMRG extrapolation is shown, with DMRG discarded weight at the top axis.

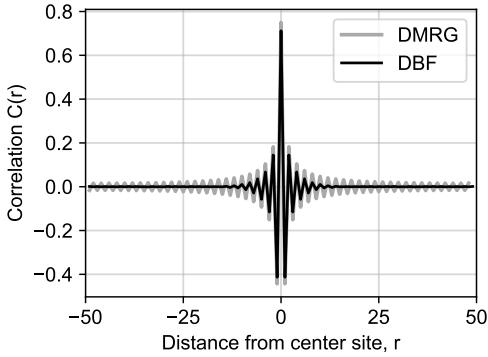


FIG. 3: Spin-spin correlation functions for 1x100 site Heisenberg lattice. $C(r) = \langle S_1 S_r \rangle - \langle S_1 \rangle \langle S_r \rangle$. Blue are for converged DMRG results. Black line is vDBF($1\text{e-}4$)

this data that a 10-fold decrease in ϵ results in a commensurate 10-fold increase in cost (len(H), cpu time).

This data already illustrates a more general feature that will be seen with the following examples as well: decreasing the threshold, ϵ , decreases truncation errors (discarded weight), but generally decreases convergence rate (hence the larger variances for smaller ϵ). However, since we have an effective way to correct for convergence errors, we are still able to obtain qualitatively correct results from relatively few iterations.

In Panel (a) in Fig. 2, the convergence and extrapolations are shown for each of the four threshold values for the 1×100 lattice. As described above, we choose our data cutoff point so as to minimize the uncertainty in our linear fit, and the shaded regions in the curve show the difference between linear and quadratic extrapolations. For this system, the differences between linear and quadratic fits are undetectable in the plots, indicating a good fit. We find that even the raw energy, $\langle 0 | H^{(i_{\text{final}})} | 0 \rangle$ without extrapolation, is already less than 1% error from the ground state energy for all 4 truncation thresholds.

To develop better physical insight into the nature of our approximations, in Fig. 3 we have computed the spin-spin correlation function vs distance, $C(r) = \langle S_1 S_r \rangle - \langle S_1 \rangle \langle S_r \rangle$. From this plot, we can see the origin of the computational speedups. DMRG shows the expected polynomial decay of $C(r)$, with non-trivial alternating correlations extending across the entire 100-site chain. In contrast, the vDBF correlations decay exponentially. Since the sequence of Pauli rotations is chosen to greedily minimize the energy, non-local correlations that don't directly affect the energy are naturally discarded. This also indicates that for gapped systems away from critical points (e.g., introducing anisotropy in J), we would expect vDBF to perform even better, since there the correlations also decay exponentially.

2. 6×6 Heisenberg

Considering next a two-dimensional system that is small enough for DMRG to be easily converged, we provide the site energies and convergence data in Table I, as well as extrapolation plots in Fig. 2(b). As expected, the DMRG calculations required a higher bond dimension and greater CPU time despite the smaller size, due to the two-dimensional geometry. We also observe an increase in the computational cost of the vDBF calculations due to the fact that a higher percentage of operators directly contribute to the energy. Despite the smaller size, the $\epsilon = 1\text{e-}5$ calculation ended with over 16 million Pauli strings in the Hamiltonian, significantly larger than the 3 million terms in the 1×100 system. Here, we see that while vDBF is able to provide accurate results ($< 0.1\%$), the CPU time is roughly a factor of 10 times slower than the reference DMRG calculation. While parallelization might be possible to reduce the vDBF cost in the future (though this is not obvious), overall, this system is clearly better treated with DMRG.

3. 10×10 Heisenberg

Compared to the previous two examples, the 10×10 lattice is both large and high-dimensional, making direct TNS treatment challenging. In Table I, we obtained relatively good DMRG convergence with a bond dimension of 3500, which led to a very small extrapolation correction of around 0.0002 J/site (this extrapolation is also shown in Fig. 2).

Each of the four thresholds used for the vDBF calculations were able to obtain less than 1% error with respect to the extrapolated DMRG energies, all requiring significantly less resources than DMRG. For example, the $\epsilon = 1\text{e-}3$ calculation took only 10 minutes on a single thread, with an error of 0.45%, whereas the DMRG calculation took more than 2 days on 64 cores.

B. Hubbard Hamiltonian

The Fermi-Hubbard Hamiltonian:

$$H = -t \sum_{\langle ij \rangle, \sigma} \hat{a}_{i\sigma}^\dagger \hat{a}_{j\sigma} + U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \quad (20)$$

is a canonical model for describing interactions in electronic systems across a wide range of phenomenological settings. Changes in the balance of kinetic and potential energy (t/U) or connectivity ($\langle ij \rangle$) give rise to diverse physics, including metal-insulator transitions, magnetism, Mott insulating behavior, and high-temperature superconductivity. Furthermore, it serves as a fundamental benchmark for evaluating new computational methods for many-body physics.[83]

In this section, we demonstrate the ability of vDBF to estimate the ground state energy of the Hubbard model.

Hubbard Lattice	#Qubits	DMRG	vDBF				
			$\epsilon :$	1e-2	1e-3	5e-4	
1x64	128	Energy/Site: -1.062475	Energy/Site:	-1.052131(2)	-1.0577(2)	-1.06060(7)	-1.06340(1)
			Error:	0.97%	0.45%	0.18%	0.09%
			E Variance:	0.0001	0.0852	0.1636	0.3133
			Discarded W:	1.793	0.304	0.189	0.1104
			Iterations:	626	1000	1000	1000
			len(H):	4198	86654	182493	578148
			Time(min):	20.4	38.1	85.8	288.9
			Threads:	64	1	1	1
4x4	32	Energy/Site: -1.238971	Energy/Site:	-1.2358(1)	-1.24019400(2)	-1.240219(3)	-1.238313(2)
			Error:	0.26%	0.0987%	0.1007%	0.05%
			E Variance:	0.000	0.0032	0.0036	0.009
			Discarded W:	0.940	0.193	0.121	0.078
			Iterations:	243	1000	1000	1000
			len(H):	1017	29635	67973	285639
			Time(min):	17.2	15.7	31.1	206.1
			Threads:	64	1	1	1
8x8	128	Var E/Site: -1.308112 Extrap E/Site: -1.331(2)	Energy/Site:	-1.31211(2)	-1.33508(5)	-1.334071(7)	-1.338(2)
			Error:	1.42%	0.31%	0.23%	0.53%
			E Variance:	0.001	1.3017	1.6786	2.4278
			Discarded W:	4.964	1.324	0.909	0.561
			Iterations:	1000	1000	1000	1000
			len(H):	6028	160387	407269	1243355
			Time(min):	4996.0	5.0	75.2	184.9
			Threads:	64	1	1	1

TABLE II: Energy/Site values for Hubbard models. All lattices have open boundary conditions. vDBF calculations are each run with the threshold, ϵ , defined in the top row, and `max_iter`=1000 and `n_rots`=50. Uncertainties in the vDBF energies, shown in parentheses, are given by half the difference between linear and quadratic extrapolation, as described in Sec. III 0 a. “Extrap E/Site” denotes DMRG extrapolated value, with uncertainty reported as the standard error in the y -intercept of the linear fit (also shown in Fig. 4(c)).

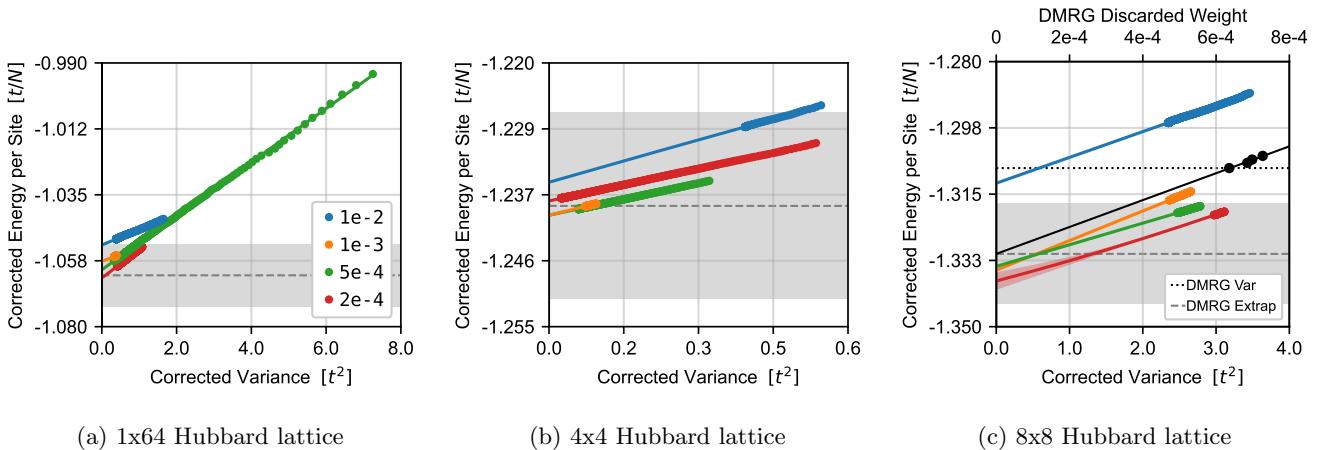


FIG. 4: Energy vs Variance extrapolations for Hubbard lattice using a series of vDBF thresholds: $\epsilon = 1\text{e-}2$ (blue), $1\text{e-}3$ (orange), $5\text{e-}4$ (green), $2\text{e-}4$ (red). Two qubits per fermionic site require 128, 32, and 128 qubits, respectively. Extrapolations shown as solid lines. Uncertainties (difference between linear and quadratic fits) are shown as shaded regions. DMRG best estimates (dashed grey line) and 1% error region w.r.t. DMRG best estimate (grey shaded area).

In the previous section, we investigated the Heisenberg model, which is equivalent to the strong-coupling limit of the Hubbard model at half-filling. In this section, we focus on the Hubbard model at moderate coupling ($t = U$) away from half-filling, targeting the true ground state of the Hamiltonian. Analogous to the Heisenberg results, we first compute a large one-dimensional 64-site lattice, then a small two-dimensional 16-site lattice for which DMRG can easily provide benchmark results. We then study a two-dimensional 64-site lattice (128 qubits) for which DMRG results were challenging to obtain.

In order to simulate this fermionic system with this Pauli operator representation, we use a Jordan-Wigner transformation to map the fermionic sites onto distinguishable lattice sites. The vDBF and DMRG energies and convergence data are presented in Table II, and the extrapolation plots are shown in Fig. 4.

1. 1×64 Hubbard

For the linear 1×64 site Hubbard model, all 4 thresholds considered achieved less than 1% error compared to the DMRG benchmark. Consistent with the Heisenberg model in the previous section, both accuracy and cost consistently increase with decreasing truncation threshold. For example, the $\epsilon = 1\text{e-}2$ calculation achieved 99% accuracy while being 10 times faster than DMRG. However, to achieve 99.9% accuracy, the vDBF($2\text{e-}4$) calculations were over 10 times slower than DMRG, illustrating the practical role that vDBF could play in providing quick estimates of ground state energies.

In Fig. 4(a), we see that for all thresholds less than $1\text{e-}2$, the accuracy of vDBF had already reached 99% even before extrapolation.

2. 4×4 Hubbard

For the 4×4 lattice, we see a notably fast convergence of the vDBF results, with all methods providing less than 1% error, even before extrapolation. For this system, we were able to converge the vDBF much more tightly, with each calculation reaching a variance of less than 0.01. This is also reflected in the plot in Fig. 4(b), where almost all the data points used in the extrapolation already fell within the 99% accuracy range. As with the 1×64 lattice, the cheaper calculations were able to obtain accurate energy estimates of around 99%-99.9% while being either faster or comparable to DMRG in terms of CPU time.

3. 8×8 Hubbard

Shifting our focus now to the large (128 qubit) two-dimensional Hubbard model, we immediately note that in Table II, the DMRG results contain both a variational energy, as well as an extrapolated energy, since we were

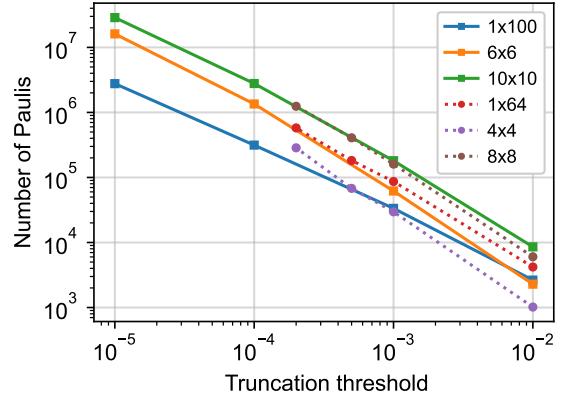


FIG. 5: Relationship between number of Pauli strings vs truncation threshold for both Heisenberg models (squares) and Hubbard models (circles).

not able to go beyond the bond dimension of 6000 while using a single compute node. As such, our DMRG extrapolation produces a rather large correction, amounting to around 2% of the extrapolated energy.

For the 3 tightest vDBF calculations, $1\text{e-}3$, $5\text{e-}4$, and $2\text{e-}4$, we observe 99% accuracy compared to the extrapolated DMRG results. However, comparing the $5\text{e-}4$ and $2\text{e-}4$ results, we notice that the error appears to increase slightly. While this may be a real consequence of the fact that the $2\text{e-}4$ calculation is slower to converge (only reaching a variance of $2.4 t^2$ in 1000 iterations), it might also be a consequence of inaccuracies in our DMRG benchmark. We find that the three tightest vDBF calculations are closer to the extrapolated than the best variational DMRG energy, even before extrapolation. As such, it's possible that the vDBF results indicate that the true ground state is somewhat lower in energy than what is predicted by DMRG.

Comparing both Heisenberg and Hubbard models, we see that the relationship between the truncation threshold, ϵ , chosen and the length of the resulting transformed Hamiltonian has a relatively consistent relationship. In Fig. 5, we show a log-log plot of the number of Paulis in the final Hamiltonian vs the truncation threshold. In all 6 systems studied, an order of magnitude decrease in the threshold leads to roughly an order of magnitude increase in size of $H^{(i)}$. We see that the Heisenberg lattice appears to have a slower operator growth than the Hubbard model (this is also reflected in the fact that we were able to obtain results with significantly tighter ϵ values for the Heisenberg examples). Finally, we also notice that both 1D examples have a slightly shallower slope, indicating a slower operator spreading due to the higher sparsity of the 1D lattices.

IV. CONCLUSIONS

In this paper, we have demonstrated that Sparse Pauli Dynamics, originally developed to simulate quantum circuits classically, can serve as an efficient computational kernel for ground state energy estimation. By combining this with a double bracket-inspired variational approach and new extrapolation techniques, we obtain accurate ground state energies (< 1% error from DMRG benchmarks) with modest computational resources, in some cases outperforming DMRG by orders of magnitude in wall time. The techniques developed to probe the boundaries of quantum advantage may thus be repurposed as practical tools for classical many-body simulation.

The most impressive performance was seen for the 2D Hubbard model, where the vDBF was able to obtain a ground state energy with less than 1% error for a 128 qubit square lattice Hubbard model in only 3 hrs on a single CPU thread (even without extrapolation), while our best DMRG variational energy had comparable accuracy ($\approx 2\%$ error) and took over 3 days on 64 threads.

While we don't claim our DMRG timings are definitive (there are likely ways one might increase the efficiency that we have not considered), these initial results indicate that circuit simulation techniques like Sparse Pauli Dynamics and PauliPropagation have a lot of promise for playing a role in future classical methods.

This work highlights several directions for future development. The optimization proceeds quickly in early iterations but slows considerably later, making full convergence difficult to achieve. Although our extrapolation techniques work well, limited non-local optimizations (e.g., variationally optimizing the last k operators rather than just the most recent one) might significantly accelerate convergence. Symmetry preservation is a challenge. In this paper, we have considered single Pauli rotations, which makes simple symmetries like particle number difficult to enforce. This likely slows convergence and complicates targeting specific states (like half-filling in the Hubbard model). Future work will explore spin and particle number conservation. As in the hybrid Heisenberg-Schrödinger picture approach of Ref. [36], one could couple vDBF with a Schrödinger picture correction to accelerate convergence; our preliminary investigations in this direction have been encouraging. We note that vDBF prioritizes energy accuracy over other observables, making characterization of ground state properties challenging. As shown in Fig. 3, we observed that spatial correlations decay exponentially rather than with the correct power law, suggesting that the method is best suited for energy estimation rather than detailed characterization of ground state properties. However, property estimation might be improved through modifications of the cost function (perhaps by adding constraints). This will be investigated in future work.

V. ACKNOWLEDGMENTS

The authors are grateful for the generous support from the National Science Foundation (Award No. 2414574). This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

REFERENCES

- [1] Møller, C. K.; Plesset, M. S. Note on an Approximation Treatment for Many-Electron Systems. *Physical Review* **1934**, *46*, 618–622.
- [2] Bartlett, R. J.; Purvis, G. D. Many-body perturbation theory, coupled-pair many-electron theory, and the importance of quadruple excitations for the correlation problem. *Int J of Quantum Chemistry* **1978**, *14*, 561–581.
- [3] Shavitt, I.; Bartlett, R. J. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*, 1st ed.; Cambridge University Press, 2009.
- [4] Bartlett, R. J.; Musial, M. Coupled-cluster theory in quantum chemistry. *Rev. Mod. Phys.* **2007**, *79*, 291–352.
- [5] Bartlett, R. J. Many-Body Perturbation Theory and Coupled Cluster Theory for Electron Correlation in Molecules. *Annual Review of Physical Chemistry* **1981**, *32*, 359–401.
- [6] Hedin, L. New Method for Calculating the One-Particle Green's Function with Application to the Electron-Gas Problem. *Phys. Rev.* **1965**, *139*, A796–A823.
- [7] Marie, A.; Ammar, A.; Loos, P.-F. *Advances in Quantum Chemistry*; Elsevier, 2024; Vol. 90; pp 157–184.
- [8] Aryasetiawan, F.; Gunnarsson, O. The GW method. *Reports on Progress in Physics* **1998**, *61*, 237.
- [9] Foulkes, W. M. C.; Mitas, L.; Needs, R. J.; Rajagopal, G. Quantum Monte Carlo simulations of solids. *Rev. Mod. Phys.* **2001**, *73*, 33–83.
- [10] Booth, G. H.; Thom, A. J. W.; Alavi, A. Fermion Monte Carlo without fixed nodes: A game of life, death, and annihilation in Slater determinant space. *The Journal of Chemical Physics* **2009**, *131*, 054106.
- [11] Cleland, D.; Booth, G. H.; Alavi, A. Communications: Survival of the fittest: Accelerating convergence in full configuration-interaction quantum Monte Carlo. *The Journal of chemical physics* **2010**, *132*.
- [12] Motta, M.; Zhang, S. Ab initio computations of molecular systems by the auxiliary-field quantum Monte Carlo method. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2018**, *8*, e1364.
- [13] Zhang, S.; Krakauer, H. Quantum Monte Carlo Method using Phase-Free Random Walks with Slater Determinants. *Phys. Rev. Lett.* **2003**, *90*, 136401.
- [14] Zhang, S.; Carlson, J.; Gubernatis, J. E. Constrained path Monte Carlo method for fermion ground states. *Phys. Rev. B* **1997**, *55*, 7464–7477.
- [15] Chan, G. K.-L.; Sharma, S. The Density Matrix Renormalization Group in Quantum Chemistry. *Annual Review of Physical Chemistry* **2011**, *62*, 465–481, PMID: 21219144.
- [16] Marti, K. H.; Reiher, M. New electron correlation theories for transition metal chemistry. *Phys. Chem. Chem.*

- Phys.* **2011**, *13*, 6750–6759.
- [17] Wouters, S.; Van Neck, D. The density matrix renormalization group for ab initio quantum chemistry. *The European Physical Journal D* **2014**, *68*, 272–272.
- [18] White, S. R. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.* **1992**, *69*, 2863–2866.
- [19] White, S. R.; Martin, R. L. *Ab initio* quantum chemistry using the density matrix renormalization group. *The Journal of Chemical Physics* **1999**, *110*, 4127–4130.
- [20] White, S. R. Density matrix renormalization group algorithms with a single center site. *Phys. Rev. B* **2005**, *72*, 180403.
- [21] Ma, Y.; Wen, J.; Ma, H. Density-matrix renormalization group algorithm with multi-level active space. *The Journal of Chemical Physics* **2015**, *143*, 034105, Publisher: American Institute of Physics.
- [22] Hyatt, K.; Stoudenmire, E. M. DMRG approach to optimizing two-dimensional tensor networks. *arXiv preprint arXiv:1908.08833* **2019**,
- [23] Fishman, M. T.; Vanderstraeten, L.; Zauner-Stauber, V.; Haegeman, J.; Verstraete, F. Faster methods for contracting infinite 2d tensor networks. *arXiv preprint arXiv:1711.05881* **2017**,
- [24] Ma, L.; Fishman, M.; Stoudenmire, E. M.; Solomonik, E. Approximate contraction of arbitrary tensor networks with a flexible and efficient density matrix algorithm. *Quantum* **2024**, *8*, 1580.
- [25] Zauner-Stauber, V.; Vanderstraeten, L.; Fishman, M. T.; Verstraete, F.; Haegeman, J. Variational optimization algorithms for uniform matrix product states. *Phys. Rev. B* **2018**, *97*, 045145.
- [26] Vidal, G. Classical simulation of infinite-size quantum lattice systems in one spatial dimension. *Physical review letters* **2007**, *98*, 070201.
- [27] Verstraete, F.; Cirac, J. I. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066* **2004**,
- [28] Stoudenmire, E. M. Learning relevant features of data with multi-scale tensor networks. *Quantum Science and Technology* **2018**, *3*, 034003.
- [29] Vidal, G. Entanglement Renormalization. *Phys. Rev. Lett.* **2007**, *99*, 220405.
- [30] Troyer, M.; Wiese, U.-J. Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations. *Physical review letters* **2005**, *94*, 170201.
- [31] Quantum error correction below the surface code threshold. *Nature* **2025**, *638*, 920–926.
- [32] Kim, Y.; Eddins, A.; Anand, S.; Wei, K. X.; van den Berg, E.; Rosenblatt, S.; Nayfeh, H.; Wu, Y.; Zalewski, M.; Temme, K.; Kandala, A. Evidence for the Utility of Quantum Computing before Fault Tolerance. *Nature* **2023**, *618*, 500–505.
- [33] Bluvstein, D.; Evered, S. J.; Geim, A. A.; Li, S. H.; Zhou, H.; Manovitz, T.; Ebadi, S.; Cain, M.; Kalinowski, M.; Hangleiter, D.; others Logical quantum processor based on reconfigurable atom arrays. *Nature* **2024**, *626*, 58–65.
- [34] Lee, S. et al. Evaluating the Evidence for Exponential Quantum Advantage in Ground-State Quantum Chemistry. *Nat Commun* **2023**, *14*, 1952.
- [35] Begušić, T.; Chan, G. K.-L. Fast Classical Simulation of Evidence for the Utility of Quantum Computing before Fault Tolerance. 2023.
- [36] Begušić, T.; Gray, J.; Chan, G. K.-L. Fast and Converged Classical Simulations of Evidence for the Utility of Quantum Computing before Fault Tolerance. *Science Advances* **2024**, *10*, eadk4321.
- [37] Tindall, J.; Fishman, M.; Stoudenmire, E. M.; Sels, D. Efficient Tensor Network Simulation of IBM’s Eagle Kicked Ising Experiment. *PRX Quantum* **2024**, *5*, 010308.
- [38] Rall, P.; Liang, D.; Cook, J.; Kretschmer, W. Simulation of Qubit Quantum Circuits via Pauli Propagation. *Phys. Rev. A* **2019**, *99*, 062337.
- [39] Rudolph, M. S.; Fontana, E.; Holmes, Z.; Cincio, L. Classical Surrogate Simulation of Quantum Systems with LOWESA. 2023.
- [40] Schuster, T.; Yin, C.; Gao, X.; Yao, N. Y. A Polynomial-Time Classical Algorithm for Noisy Quantum Circuits. 2024.
- [41] Cirstoiu, C. A Fourier Analysis Framework for Approximate Classical Simulations of Quantum Circuits. 2024.
- [42] Fontana, E.; Rudolph, M. S.; Duncan, R.; Rungger, I.; Cirstoiu, C. Classical Simulations of Noisy Variational Quantum Circuits. *npj Quantum Inf* **2025**, *11*, 84.
- [43] Begušić, T.; Hejazi, K.; Chan, G. K. Simulating Quantum Circuit Expectation Values by Clifford Perturbation Theory. *The Journal of Chemical Physics* **2025**, *162*.
- [44] Schuster, T.; Yin, C.; Gao, X.; Yao, N. Y. A Polynomial-Time Classical Algorithm for Noisy Quantum Circuits. *Phys. Rev. X* **2025**, *15*, 041018.
- [45] Angrisani, A.; Schmidhuber, A.; Rudolph, M. S.; Cerezo, M.; Holmes, Z.; Huang, H.-Y. Classically Estimating Observables of Noiseless Quantum Circuits. 2025.
- [46] Angrisani, A.; Mele, A. A.; Rudolph, M. S.; Cerezo, M.; Holmes, Z. Simulating Quantum Circuits with Arbitrary Local Noise Using Pauli Propagation. 2025.
- [47] Miller, A.; Holmes, Z.; Salehi, Ö.; Chakraborty, R.; Nykänen, A.; Zimborás, Z.; Glos, A.; García-Pérez, G. Simulation of Fermionic Circuits Using Majorana Propagation. 2025.
- [48] Robbiati, M. et al. Double-Bracket Quantum Algorithms for High-Fidelity Ground State Preparation. 2025.
- [49] Rudolph, M. S.; Jones, T.; Teng, Y.; Angrisani, A.; Holmes, Z. Pauli Propagation: A Computational Framework for Simulating Quantum Systems. 2025.
- [50] González-García, G.; Cirac, J. I.; Trivedi, R. Pauli Path Simulations of Noisy Quantum Circuits beyond Average Case. *Quantum* **2025**, *9*, 1730.
- [51] While one can always find a change of basis that makes the ground state sparse (e.g., the eigenbasis), this is computationally intractable, whereas Hamiltonians are generally sparse in computationally tractable bases (e.g., Pauli, or fermionic operator bases).
- [52] Aguilar, G.; Cichy, S.; Eisert, J.; Bittel, L. Full Classification of Pauli Lie Algebras. 2024.
- [53] Kökcü, E.; Steckmann, T.; Wang, Y.; Freericks, J. K.; Dumitrescu, E. F.; Kemper, A. F. Fixed Depth Hamiltonian Simulation via Cartan Decomposition. *Phys. Rev. Lett.* **2022**, *129*, 070501.
- [54] Wiersema, R.; Kökcü, E.; Kemper, A. F.; Bakalov, B. N. Classification of Dynamical Lie Algebras of 2-Local Spin Systems on Linear, Circular and Fully Connected Topologies. *npj Quantum Inf* **2024**, *10*, 110.
- [55] Magoulas, I.; Evangelista, F. A. Clifford Transformations for Fermionic Quantum Systems: From Paulis to Majors

- ranas to Fermions. 2025.
- [56] Jayakumar, P.; Zeng, T.; Izmaylov, A. F. On the Feasibility of Exact Unitary Transformations for Many-body Hamiltonians. 2025.
- [57] Evangelista, F. A.; Magoulas, I. Exact Closed-Form Unitary Transformations of Fermionic Operators. *Phys. Rev. A* **2025**, *111*, 042825.
- [58] Martinez, V.; Angrisani, A.; Pankovets, E.; Fawzi, O.; Fran  a, D. S. Efficient Simulation of Parametrized Quantum Circuits under Non-Unital Noise through Pauli Backpropagation. *Phys. Rev. Lett.* **2025**, *134*, 250602.
- [59] Glazek, S. D.; Wilson, K. G. Perturbative Renormalization Group for Hamiltonians. *Phys. Rev. D* **1994**, *49*, 4214–4218.
- [60] Glazek, S. D.; Wilson, K. G. Renormalization of Hamiltonians. *Phys. Rev. D* **1993**, *48*, 5863–5872.
- [61] Wegner, F. Flow-Equations for Hamiltonians. *Annalen der Physik* **1994**, *506*, 77–91.
- [62] Hergert, H.; Bogner, S. K.; Lietz, J. G.; Morris, T. D.; Novario, S. J.; Parzuchowski, N. M.; Yuan, F. In *An Advanced Course in Computational Nuclear Physics: Bridging the Scales from Quarks to Neutron Stars*; Hjorth-Jensen, M., Lombardo, M. P., van Kolck, U., Eds.; Springer International Publishing: Cham, 2017; pp 477–570.
- [63] Evangelista, F. A. A Driven Similarity Renormalization Group Approach to Quantum Many-Body Problems. *The Journal of chemical physics* **2014**, *141*, 054109–054109.
- [64] Marie, A.; Loos, P.-F. A Similarity Renormalization Group Approach to Green's Function Methods. *J. Chem. Theory Comput.* **2023**, *19*, 3943–3957.
- [65] Wang, S.; Li, C.; Evangelista, F. A. Analytic Energy Gradients for the Driven Similarity Renormalization Group Multireference Second-Order Perturbation Theory. *J. Chem. Theory Comput.* **2021**, *17*, 7666–7681.
- [66] Gluza, M. Double-Bracket Quantum Algorithms for Diagonalization. *Quantum* **2024**, *8*, 1316.
- [67] Any computational basis state can be rotated into $|0\rangle$, folding the resulting unitary into $H(0)$.
- [68] Grimsley, H. R.; Economou, S. E.; Barnes, E.; Mayhall, N. J. An Adaptive Variational Algorithm for Exact Molecular Simulations on a Quantum Computer. *Nat. Commun.* **2019**, *10*, 3007.
- [69] Tang, H. L.; Shkolnikov, V.; Barron, G. S.; Grimsley, H. R.; Mayhall, N. J.; Barnes, E.; Economou, S. E. Qubit-ADAPT-VQE: An Adaptive Algorithm for Constructing Hardware-Efficient Ansatzes on a Quantum Processor. *PRX Quantum* **2021**, *2*, 020310.
- [70] We note that additional applications of the CLIP function are applied at various points to improve efficiency, but are not included in Algorithm 1 for clarity. For example, after forming the flow generator, G , this operator is clipped. However, this is clipped with a tight threshold (default 10^{-6}) so as to not affect the accuracy.
- [71] Ryabinkin, I. G.; Lang, R. A.; Genin, S. N.; Izmaylov, A. F. Iterative Qubit Coupled Cluster Approach with Efficient Screening of Generators. *J. Chem. Theory Comput.* **2020**, *16*, 1055–1063.
- [72] Ryabinkin, I. G.; Yen, T.-C.; Genin, S. N.; Izmaylov, A. F. Qubit Coupled-Cluster Method: A Systematic Approach to Quantum Chemistry on a Quantum Computer. *J. Chem. Theory Comput.* **2018**, *14*, 6317.
- [73] Lang, R. A.; Mehendale, S. G.; Ryabinkin, I. G.; Izmaylov, A. F. Multistate Iterative Qubit Coupled Cluster (MS-iQCC): A Quantum-Inspired, State-Averaged Approach to Ground- and Excited-State Energies. 2025.
- [74] Mizusaki, T.; Imada, M. Precise estimation of shell model energy by second-order extrapolation method. *Phys. Rev. C* **2003**, *67*, 041301.
- [75] Shimizu, N.; Utsuno, Y.; Mizusaki, T.; Otsuka, T.; Abe, T.; Honma, M. Novel extrapolation method in the Monte Carlo shell model. *Physical Review C—Nuclear Physics* **2010**, *82*, 061305.
- [76] Shimizu, N.; Utsuno, Y.; Mizusaki, T.; Otsuka, T.; Abe, T.; Honma, M. Extrapolation method in the Monte Carlo Shell Model and its applications. *AIP Conference Proceedings* **2011**, *1355*, 138–144.
- [77] Mizusaki, T.; Shimizu, N. New variational Monte Carlo method with energy variance extrapolation for large-scale shell-model calculations. *Phys. Rev. C* **2012**, *85*, 021301.
- [78] nmayhall Nmayhall/PauliOperators.Jl. 2025.
- [79] Fishman, M.; White, S. R.; Stoudenmire, E. M. The ITensor Software Library for Tensor Network Calculations. <https://arxiv.org/abs/2007.14822v2>, 2020.
- [80] Alkabetz, R.; Arad, I. Tensor Networks Contraction and the Belief Propagation Algorithm. *Phys. Rev. Res.* **2021**, *3*, 023073.
- [81] Evenbly, G.; Pancotti, N.; Milsted, A.; Gray, J.; Chan, G. K.-L. Loop Series Expansions for Tensor Networks. <https://arxiv.org/abs/2409.03108v2>, 2024.
- [82] Tindall, J.; Fishman, M. T. Gauging Tensor Networks with Belief Propagation. <https://arxiv.org/abs/2306.17837v5>, 2023.
- [83] Simons Collaboration on the Many-Electron Problem et al. Solutions of the Two-Dimensional Hubbard Model: Benchmarks and Results from a Wide Range of Numerical Algorithms. *Phys. Rev. X* **2015**, *5*, 041041.

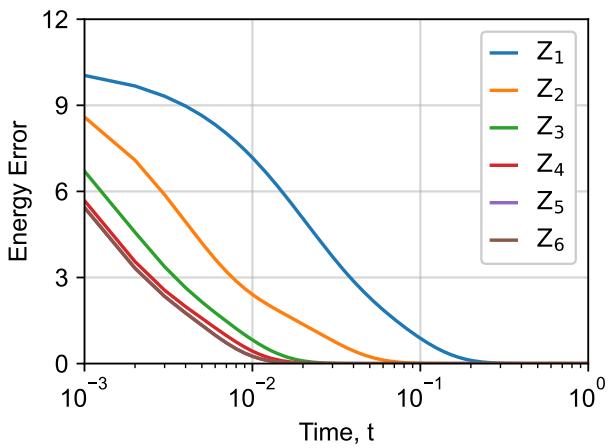


FIG. 6: Convergence of the exact continuous double bracket flow for different approximations to the projector for a 1x6 Heisenberg lattice. Z_i indicates that only terms up to i are kept in the projector.

Appendix A: Comparison to exact Projector

In the theory section above, we motivated the use of the simple sum of single qubit Z operators in the flow

generator, $[H(s), \sum_i Z_i]$ as an approximation to the more complicated projector-based generator, $[H(s), |\psi\rangle\langle\psi|]$. While this approximation was made for computational considerations, we show in Fig. 6 that this seemingly aggressive approximation still works to drive the system to the ground state. We plot the difference between the exact ground state energy, and the time-dependent expectation value of $H(s)$, for various approximations to the exact projection-based generator for a small system of a one-dimensional 6-site Heisenberg model. We see that while the full projector (Z_6) exhibits the fastest convergence (Z_5 is actually identical here, since Z_6 simply adds the $ZZZZZZ$ operator, which is a symmetry), all the other n -body approximations still eventually converge.

While the Z_1 approximation is the slowest, the implementation is much more efficient. Further, in our final algorithm, the final step-size is not determined by directly integrating the double bracket flow, but rather chosen to variationally minimize the energy.