

# Classical optimization algorithms for diagonalizing quantum Hamiltonians

Taehee Ko,<sup>1,\*</sup> Sangkook Choi,<sup>1</sup> Hyowon Park,<sup>2</sup> and Xiantao Li<sup>3,†</sup>

<sup>1</sup>*School of Computational Sciences, Korea Institute for Advanced Study*

<sup>2</sup>*Department of Physics, University of Illinois at Chicago*

<sup>3</sup>*Department of Mathematics, the Pennsylvania State University.*

Diagonalizing a Hamiltonian, which is essential for simulating its long-time dynamics, is a key primitive in quantum computing and has been proven to yield a quantum advantage for several specific families of Hamiltonians. Yet, despite its importance, only a handful of diagonalization algorithms exist, and correspondingly few families of fast-forwardable Hamiltonians have been identified. This paper introduces classical optimization algorithms for Hamiltonian diagonalization by formulating a cost function that penalizes off-diagonal terms and enforces unitarity via an orthogonality constraint, both expressed in the Pauli operator basis. We show that the landscape is benign: every stationary point is a global minimum, and any non-trivial stationary point yields a valid diagonalization, eliminating suboptimal solutions. Under a mild, numerically validated non-convexity assumption, we prove that the proposed optimization algorithm converges. In addition, we derive an *a posteriori* error bound that converts the optimization error directly into a bound on the Hamiltonian’s diagonalization accuracy. We pinpoint a class of Hamiltonians that highlights severe drawbacks of existing methods, including exponential per-iteration cost, exponential circuit depth, or convergence to spurious optima. Our approach overcomes these shortcomings, achieving polynomial-time efficiency while provably avoiding suboptimal points. As a result, we broaden the known realm of fast-forwardable systems, showing that quantum-diagonalizable Hamiltonians extend to cases generated by exponentially large Lie algebras. On the practical side, we also present a randomized-coordinate variant that achieves a more efficient per-iteration cost than the deterministic counterpart. We demonstrate the effectiveness of these algorithms through explicit examples and numerical experiments.

## I. INTRODUCTION

With a provable computational advantage for short-time Hamiltonian simulation [23], a variety of numerical schemes have been developed, including those based on the Trotter formula [5, 6, 28, 35], quantum signal processing (QSP) [24], the quantum singular value transformation (QSVT) [13], and Taylor-series methods [3]. While these schemes achieve optimal or near-optimal complexity for many Hamiltonians, the circuit complexity still grows at least linearly with the simulation time  $T$ , as compatible with the no-fast-forwarding theorem [2]. Consequently, practical tasks, such as long-time simulations and high-precision energy measurements, remain out of reach.

This limitation has motivated efforts to identify Hamiltonians whose dynamics can be fast-forwarded or simulated with circuit complexity that is sublinear in  $T$  and polynomial in the number of qubits  $n$ . Such treatment is often regarded as efficient diagonalization. Notable examples include Hamiltonians with quadratic fermionic operators and commuting local terms [2] as well as their extensions involving polynomially sized Lie algebras [16, 17, 27], certain spin Hamiltonians [21], and Hamiltonians admitting efficient quantum-circuit diagonalizations [2, 26].

---

\* [kthmomo@kias.re.kr](mailto:kthmomo@kias.re.kr)

† [xxl12@psu.edu](mailto:xxl12@psu.edu)

Together with the identification of efficiently-diagonalizable Hamiltonians, significant efforts have been dedicated to developing diagonalization algorithms. The primary objective of those algorithms is to output a quantum circuit that encodes a unitary corresponding to the resulting decomposition. Notable examples include eigen-decomposition [8, 14] and Cartan-decomposition [7, 21, 32, 38]. Importantly, these algorithms can serve as a quantum compiler for more coherent Hamiltonian simulation than direct Trotterization. For instance, Somma [32] introduced a Lie-algebraic diagonalization algorithm, reminiscent of Jacobi’s method, to prepare generalized coherent states. Kökcu et al. [21] proposed an optimization routine to compute a Cartan decomposition and demonstrated long-time Hamiltonian simulations for certain models. An improved, recursive Cartan decomposition method [38] offers computationally-efficient unitary synthesis. In addition to these classical algorithms, a variational quantum algorithm by Cirstoiu et al. [8], which optimizes a parameterized circuit to approximate an eigen-decomposition, showed Hamiltonian simulations longer than the Trotter method. As anticipated, long-time Hamiltonian simulations can facilitate the investigation of quantum many-body dynamics across a range of models, including time-dependent Hamiltonians, such as excitation dynamics with short-time external fields [36] and various spin chain dynamics models [37].

Despite these advances, significant computational challenges still persist in terms of compiling a unitary. For example, the Lie-diagonalization [16, 32] requires a growing circuit depth to be inversely proportional to the target precision  $\epsilon$  and scales linearly with the Lie algebra’s dimension. By contrast, the method in [21] achieves fixed-depth circuits scaling at most polynomially for systems of polynomially sized Lie algebras, such as the transverse-field XY (TFTY) model. However, evaluating their cost function may incur exponential overhead due to exponentially many coefficient calculations: For the example of the TFTXY model, it leads to  $\mathcal{O}(2^{n^2})$  complexity. The double-bracket approach in [14] in principle applies to more general Hamiltonians, but only a small number of iterations are allowed due to the fact that the circuit depth scales exponentially with the number of iterations. Variational methods [8], on the other hand, rely on nonconvex optimization, and may converge to local minima or saddle points, and thus might be unreliable for long-time dynamics or large systems.

In this work, we introduce a new framework for Hamiltonian diagonalization. We first define a cost function, via Pauli strings, to encode both off-diagonal terms and the orthogonality constraints. We then develop a gradient-descent method with an explicit normalization step to avoid trivial solutions. Notably, the per-iteration cost of our randomized variant scales quadratically with respect to the dimension of the Hilbert space, which is cheaper than the Lie-algebraic approaches [21, 32] in general. This framework, unlike the variational approaches, avoids suboptimal solutions, as we prove that any nonzero stationary point of the cost landscape is a global minimum, corresponding to a valid diagonalization. Remarkably, if a proper Lie algebra associated to Hamiltonian is known in advance (as in [21]), our formulation reduces to that setting, yet our parameterization remains more efficient. Under a mild nonconvexity condition, we establish sublinear or linear convergence guarantees and demonstrate that approximate cost minimization yields accurate diagonalizations of  $H$ . We test both deterministic and randomized optimization algorithms in several numerical experiments to demonstrate the efficiency of our approach. Finally, we identify a family of Hamiltonians that pose substantial challenges for existing algorithms, e.g., incurring exponential per-iteration costs, requiring deep circuits, or converging to suboptimal solutions, while our algorithm maintains polynomial efficiency and converges to a global minimum, given a known unitary decomposition.

The rest of the paper is organized as follows. We first present an optimization formulation by introducing the cost function for Hamiltonian diagonalization. Section III analyzes the properties of this cost function from an optimization perspective and studies both deterministic and randomized algorithms. In Section IV, we show Hamiltonians that incur exponential per-iteration cost under

existing Lie-algebraic methods [21, 32] but only polynomial cost with our algorithm. In Section V, we present numerical results for several representative models.

## II. METHOD DESCRIPTION

Consider an  $n$ -qubit Hamiltonian of the form

$$H = \sum_{j=1}^M c_j Q_j, \quad (1)$$

where  $c_j \in \mathbb{R}$  and  $Q_j$  denotes a Pauli string, i.e., a tensor product of  $n$  Pauli matrices  $Q_j = \otimes_{i=1}^n \sigma_j^{(i)}$  with  $\sigma_j^{(i)} \in \{I, X, Y, Z\}$ . Many Hamiltonians in chemistry and physics can be mapped to this form, and they satisfy the  $M = \mathcal{O}(\text{poly}(n))$  count.

To diagonalize such Hamiltonians, thus enabling more efficient Hamiltonian simulations, we formulate an optimization problem. In what follows, we seek a unitary  $K$  and a diagonal matrix  $D$  such that

$$D = K^\dagger H K,$$

with the obvious, but important implication that  $e^{-itH} = K e^{-itD} K^\dagger$  can be implemented by a circuit whose depth is independent of  $t$ .

We seek such diagonalization by optimization. Before defining a cost function, we first recall that an expansion of a *general* Hermitian matrix  $A$  in terms of Pauli strings can be divided into diagonal and off-diagonal terms,

$$A = \sum_{j=1}^{4^n} a_j P_j = \sum_{\forall i \in [n], \sigma_j^{(i)} \in \{I, Z\}} a_j P_j + \sum_{\exists i \in [n], \sigma_j^{(i)} \in \{X, Y\}} a_j P_j, \quad (2)$$

where  $a_j = \frac{1}{2^n} \text{tr}(AP_j)$ . This expansion is derived from the fact that the set of Pauli strings forms an orthogonal basis of the set of matrices of size  $2^n \times 2^n$  with respect to the Hilbert-Schmidt (HS) norm.

Now, we consider a parameterized matrix  $A = K^\dagger H K$  with  $H$  being the given Hamiltonian and  $K$  being parameterized. We define a cost function as follows

$$f(\mathbf{r}, \boldsymbol{\theta}) := \sum_{P \in G_1} \text{tr} (K(\mathbf{r}, \boldsymbol{\theta})^\dagger H K(\mathbf{r}, \boldsymbol{\theta}) P)^2. \quad (3)$$

Here  $G_1$ , as a subset of the set of off-diagonal Pauli strings

$$G_1 := \{\otimes_{i=1}^n \sigma^{(i)} : \sigma^{(i)} \in \{I, X, Y, Z\} \text{ and } \exists i \in [n], \sigma^{(i)} \in \{X, Y\}\}$$

consisting of Pauli strings constituting the Pauli expansion of  $K^\dagger H K$ . In Eq. (3), we consider a parametric form of  $K$ , given by,

$$K(\mathbf{r}, \boldsymbol{\theta}) = \sum_{j=1}^d r_j e^{i\theta_j} P_j, \quad \mathbf{r} = (r_1, \dots, r_d) \in \mathbb{R}^d, \quad \boldsymbol{\theta} = (\theta_1, \dots, \theta_d) \in \mathbb{R}^d. \quad (4)$$

In our ansatz, the parameters  $\{r_j e^{i\theta_j}\}$  are exactly the Pauli-basis coefficients of  $K$  written in polar decomposition form. The integer  $d$  is the total number of Pauli strings in the ansatz (and  $2d$  is the parameter dimension) and it depends on the Hamiltonian under consideration. This definition implies that for any non-zero  $\mathbf{r}$  such that  $f(\mathbf{r}, \boldsymbol{\theta}) = 0$ , we obtain a diagonal matrix  $K(\mathbf{r}, \boldsymbol{\theta})^\dagger H K(\mathbf{r}, \boldsymbol{\theta})$ .

Notice that  $K$  in Eq. (4) may not be necessarily unitary, unless we impose an orthogonality constraint on the cost function (3). A natural constraint is the Hilbert-Schmidt norm error between  $K^\dagger K$  and the identity  $I$ ,  $\|K^\dagger K - I\|_F^2$ . Toward this end, we write the expansion of  $K^\dagger K$  with (4) in Pauli strings,

$$K^\dagger K = \sum_{P=P_i P_j, i,j \in [d]} \phi_P(\mathbf{r}, \boldsymbol{\theta}) P, \quad (5)$$

where  $\phi_P = \frac{1}{2^n} \text{tr}(K^\dagger K P)$ . Together, we define a total cost function with  $2d$  parameters,

$$F(\mathbf{r}, \boldsymbol{\theta}) := f(\mathbf{r}, \boldsymbol{\theta}) + \sum_{P \in G_2} \phi_P(\mathbf{r}, \boldsymbol{\theta})^2. \quad (6)$$

Here  $G_2 = \{P : P = P_i P_j, i, j \in [d] \text{ and } P \neq I\}$ . From (3), notice that the cost function (6) is represented by a sum of  $\mathcal{O}(d^2 M)$  terms in the worst-case scenario. We also note that the first part of this function accounts for the off-diagonal terms and the second part for the orthogonality condition. An immediate observation is that a non-zero global minimum ( $\mathbf{r} \neq 0$ ) of the cost function in (6) always exists (e.g. an eigen-decomposition of  $H$ ). In fact, we will show that any stationary point of the cost function (6) is a global minimum, and therefore any non-zero stationary point corresponds to a diagonalization unitary  $K$  and an eigen-decomposition of  $H$ .

### III. MAIN RESULT

In this section, we first present several results regarding the properties of the optimization problem (3). Then, we propose a non-convex optimization algorithm to achieve efficient diagonalization. In addition, we provide perturbation results that quantify the robustness of approximate solutions of our optimization problem Eq. (6). Our first main result states the main property of the optimization problem (6),

**Theorem 1.** *Any non-zero stationary point of the cost function (6) corresponds to an eigen-decomposition of the Hamiltonian.*

Next, we outline several properties of the optimization problem that lead to this benign property of the landscape.

#### A. Optimization landscape with all stationary points being global minima

The Lie-diagonalization algorithm in [21] satisfies the property that any stationary point of their cost function yields a Cartan decomposition of  $H$ . We show that our cost function (6) has a similar property. More specifically, we prove that any non-zero stationary point of the optimization (6) yields a unitary  $K$  and a diagonal  $K^\dagger H K$ . We refer to a non-zero point as a point  $(\mathbf{r}, \boldsymbol{\theta})$  with  $\mathbf{r} \neq 0$ .

To proceed, we first work with the cost function (3), and show that for any of its non-zero stationary points, we obtain a diagonal matrix  $K^\dagger HK$ . As a preliminary, we derive an exact formula for the partial derivatives of the cost function (3) as follows,

$$\begin{aligned}\frac{\partial f}{\partial r_j} &= 4 \sum_{P \in G_1} \text{tr}(K(\mathbf{r}, \boldsymbol{\theta})^\dagger HK(\mathbf{r}, \boldsymbol{\theta})P) \text{Re}(\text{tr}(HK(\mathbf{r}, \boldsymbol{\theta})Pe^{-i\theta_j} P_j)), \\ \frac{\partial f}{\partial \theta_j} &= -4 \sum_{P \in G_1} \text{tr}(K(\mathbf{r}, \boldsymbol{\theta})^\dagger HK(\mathbf{r}, \boldsymbol{\theta})P) \text{Im}(\text{tr}(HK(\mathbf{r}, \boldsymbol{\theta})Pr_j e^{-i\theta_j} P_j)).\end{aligned}\quad (7)$$

**Theorem 2.** *Any non-zero stationary point of the cost function  $f$  in (3),  $(\mathbf{r}_c, \boldsymbol{\theta}_c)$  with  $\mathbf{r}_c \neq 0$ , is a global minimum.*

*Proof.* Let  $(\mathbf{r}_c, \boldsymbol{\theta}_c)$  be a non-zero stationary point, that is  $\mathbf{r}_c \neq 0$ . Denote  $K_c = K(\mathbf{r}_c, \boldsymbol{\theta}_c)$ . By (7), it follows that at the stationary point,

$$0 = \frac{\partial f}{\partial r_j} = 4 \sum_{P \in G_1} \text{tr}(K_c^\dagger HK_c P) \text{Re}(\text{tr}(HK_c Pe^{-i\theta_j} P_j)) =: 4\text{Re}(\text{tr}(HK_c Be^{-i\theta_j} P_j)), \quad (8)$$

where we have defined the matrix  $B$

$$B := \sum_{P \in G_1} \text{tr}(K_c^\dagger HK_c P) P. \quad (9)$$

Note that  $B$  is Hermitian. By multiplying (8) by  $r_j$ , summing over  $j$ 's, and using the definition of  $K$  in (3), we arrive at

$$0 = \text{Re}(\text{tr}(K_c^\dagger HK_c B)) = \text{Re} \left( \sum_{P \in G_1} \text{tr}(K_c^\dagger HK_c P)^2 \right) = \sum_{P \in G_1} \text{tr}(K_c^\dagger HK_c P)^2 = f(\mathbf{r}_c, \boldsymbol{\theta}_c), \quad (10)$$

by definition (9). Therefore, this completes the proof.  $\square$

According to Theorem 2, any non-zero stationary point  $K_c$  of the cost function (3) provides us with a diagonal  $K_c^\dagger HK_c$ . However, this does not mean that  $K_c$  is unitary. To eliminate such unphysical stationary points, we impose a constraint on the optimization problem Eq. (3). As shown earlier in (6), we introduce an orthogonality constraint as follows,

$$K^\dagger K = I. \quad (11)$$

This constraint can be reformulated in terms of parameters  $\mathbf{r}$  and  $\boldsymbol{\theta}$  as follows,

**Lemma 1.** *The condition Eq. (11) is equivalent to the following constraint,*

$$\phi_P(\mathbf{r}, \boldsymbol{\theta}) := \sum_{j=1}^d c_{j,P} r_j r_{j_P} \exp(i(\theta_j - \theta_{j_P})) = \begin{cases} 1, & j = j_P, c_{j,P} = 1 \ (P = I) \\ 0, & j \neq j_P \ (P \neq I) \end{cases}, \quad (12)$$

where  $j_P$  is the index corresponding to  $j$  such that  $P_{j_P} P_j = c_{j,P} P$  with  $c_{j,P} \in \{\pm 1, \pm i\}$ .

*Proof.* By the definition of  $K$  in (3), we consider the Pauli basis expansion of  $K^\dagger K$  and check that

$$I = K^\dagger K = \left( \sum_{j=1}^d r_j e^{-i\theta_j} P_j \right) \left( \sum_{j=1}^d r_j e^{i\theta_j} P_j \right) = \sum_{P=P_i P_j, i,j \in [d]} \phi_P(\mathbf{r}, \boldsymbol{\theta}) P, \quad (13)$$

where  $\phi_P$  is defined in the statement. On the right hand side of this equation, the only remaining coefficient corresponds to the identity matrix, and the others are zeros, which recovers the constraint in the statement.  $\square$

With this constraint, we improve Theorem 2 to guarantee that the resulting optimal  $K_c$  is unitary and  $K_c^\dagger H K_c$  is diagonal.

**Theorem 3.** *Any non-zero stationary point of the total cost function in (6),  $(\mathbf{r}_c, \boldsymbol{\theta}_c)$  with  $\mathbf{r}_c \neq 0$ , is a global minimum. In addition, the matrix  $K(\mathbf{r}_c, \boldsymbol{\theta}_c)$  is unitary up to a scaling factor, and consequently,*

$$H = \frac{1}{\|\mathbf{r}\|^4} K(\mathbf{r}_c, \boldsymbol{\theta}_c) h(\mathbf{r}_c, \boldsymbol{\theta}_c) K(\mathbf{r}_c, \boldsymbol{\theta}_c)^\dagger, \quad (14)$$

thereby yielding a KHK decomposition of  $H$ .

The proof of this theorem is shown in Appendix A. From the proof, we immediately see a property of the cost function (6).

**Proposition 1.** *The cost function in (6) satisfies that*

$$\sum_j r_j \frac{\partial F}{\partial r_j} = 4F(\mathbf{r}, \boldsymbol{\theta}). \quad (15)$$

*Proof.* From (A1) and (A2), we see that the variables  $\boldsymbol{\theta}$  do not affect the scaling of  $\mathbf{r}$  and so the proof is straightforward.  $\square$

From Theorem 3, minimizing the cost function in Eq. (6) produces a unitary  $K$  up to scaling and a KHK decomposition of the Hamiltonian  $H$ . However, we should avoid the trivial solution (e.g.  $\mathbf{r} = 0$ ). Namely, the desired optimization task is formulated as,

$$\min_{\mathbf{r} \neq 0} F(\mathbf{r}, \boldsymbol{\theta}). \quad (16)$$

A simple observation is the self-similar property of the total cost function:  $F(s\mathbf{r}, \boldsymbol{\theta}) = s^4 F(\mathbf{r}, \boldsymbol{\theta})$  for any  $s \in \mathbb{R}$ . By this property and the fact that  $\min_{\mathbf{r} \neq 0} F(\mathbf{r}, \boldsymbol{\theta}) = 0$ , the optimization problem can be equivalently reformulated as a constrained optimization on the unit sphere, while removing zero stationary points,

$$\min_{\mathbf{r} \neq 0} F(\mathbf{r}, \boldsymbol{\theta}) = \min_{\|\mathbf{r}\|=1} F(\mathbf{r}, \boldsymbol{\theta}). \quad (17)$$

## B. Optimization Algorithm based on the gradient descent

In this section, we discuss optimization algorithms for solving the problem (17), and present its convergence guarantee under a mild non-convex condition. Specifically, the algorithm updates the parameters  $(\mathbf{r}, \boldsymbol{\theta})$  at each iteration: At each iteration, the algorithm applies gradient descent (GD), followed by a normalization of the parameter  $\mathbf{r}$  (in order to avoid the trivial solution  $\mathbf{r} = 0$ ). The simple algorithm is outlined in Algorithm 1. For convenience, we introduce the following notation in Algorithm 1,

$$\mathbf{x}_t := (\mathbf{r}_t, \boldsymbol{\theta}_t), \quad \mathbf{r}(\mathbf{x}_t) = \mathbf{r}_t, \quad \boldsymbol{\theta}(\mathbf{x}_t) = \boldsymbol{\theta}_t. \quad (18)$$

---

### Algorithm 1: Deterministic optimization algorithm

---

**Data:** initial guess  $\mathbf{x}_0 = (\mathbf{r}_0, \boldsymbol{\theta}_0)$  with  $\|\mathbf{r}_0\| = 1$ , learning rate schedule  $a_t \in (0, 1)$ , the maximal iteration number  $T$ , period  $p$   
**Result:** approximate unitary  $K(\mathbf{r}_T, \boldsymbol{\theta}_T)$  for a KHK decomposition of  $H$

```

for  $t = 0 : T$  do
    Construct  $\nabla F(\mathbf{x}_t)$  from (6);
     $\mathbf{y}_t = \mathbf{x}_t - a_t \nabla F(\mathbf{x}_t)$ ;
     $\mathbf{x}_{t+1} = \left( \frac{\mathbf{r}(\mathbf{y}_t)}{\|\mathbf{r}(\mathbf{y}_t)\|}, \boldsymbol{\theta}(\mathbf{y}_t) \right)$ ;
end

```

---

On the other hand, unlike the conventional GD method, Algorithm 1 involves the step of normalization, and its convergence analysis should take into account the effect of normalization. We found through our analysis a similar convergence result to that of non-convex optimization using GD under a similar non-convex condition. Specifically, we use a weaker version of the Polyak–Lojasiewicz (PL) condition that has been widely used in analyzing machine learning algorithms. A related, but more general, condition is the KL condition [22, Definition 2.3], which is a local condition and holds for any continuously-differentiable lower-bounded function. We slightly modify this condition such that the condition holds uniformly near global minima, which is stronger than the original condition. Nevertheless, we have validated the proposed condition with extensive numerical results. With this experience, we propose a non-convex condition as follows,

**Assumption 1 (Uniform KL condition).** *A function  $F(\mathbf{x})$  satisfies a uniform KL condition if there exist constants  $\delta_f > 0$ ,  $\mu > 0$  and  $\alpha \in [0, 2)$  such that*

$$\|\nabla F(\mathbf{x})\|^2 \geq 4\mu F(\mathbf{x})^\alpha, \quad (19)$$

whenever  $F(\mathbf{x}) < \delta_f$ .

Under this assumption, we show that Algorithm 1 converges to a non-trivial solution of the cost function in Eq. (17) sublinearly or linearly with respect to  $\epsilon$ , depending on the exponent  $\alpha$  determined by the given Hamiltonian and the initial guess.

**Theorem 4.** *Under Assumption 1, for any precision  $\epsilon > 0$  and a good initialization  $\mathbf{x}_0$  such that  $F(\mathbf{x}_0) < \delta_f$ , Algorithm 1 converges to a solution of the optimization problem (17) to precision  $\epsilon$  sublinearly (if  $\alpha \in (1, 2)$ ) or linearly (if  $\alpha \in [0, 1]$ ) with the number of iterations at most,*

$$T = \mathcal{O} \left( \frac{1}{\epsilon^{\max\{\alpha-1, 0\}}} \log \frac{1}{\epsilon} \right). \quad (20)$$

The proof of this theorem is shown in Appendix A.

### C. Sensitivity analysis for the approximate solution

Once a numerical solution is obtained from Algorithm 1 with high accuracy, we show that a small optimization error implies a good diagonalization of  $H$ . We provide such a posterior error bound as in the following lemma,

**Lemma 2** (A posterior error bound). *For  $(\mathbf{r}, \boldsymbol{\theta})$  obtained from Algorithm 1, there exist a diagonal matrix  $h_0$  and a off-diagonal matrix  $\Delta \in \mathbb{C}^{2^n \times 2^n}$  such that*

$$K(\mathbf{r}, \boldsymbol{\theta})^\dagger H K(\mathbf{r}, \boldsymbol{\theta}) = h_0 + \Delta, \quad (21)$$

with  $\|\Delta\|_F < \sqrt{\frac{F(\mathbf{r}, \boldsymbol{\theta})}{2^n}}$ . Furthermore, if  $\|\mathbf{r}\| = 1$  and the orthogonality constraint of  $F(\mathbf{r}, \boldsymbol{\theta})$  in (6) is less than  $\frac{\epsilon}{2^n}$  with  $\epsilon \leq \frac{1}{4}$ , then, the following bound holds,

$$\left\| H - \tilde{H} \right\|_2 \leq \frac{F(\mathbf{r}, \boldsymbol{\theta})}{2^{n-1}} + 6(1 + \sqrt{F(\mathbf{r}, \boldsymbol{\theta})})\|H\|_F\sqrt{\epsilon}, \quad \tilde{H} := K(\mathbf{r}, \boldsymbol{\theta})h_0K(\mathbf{r}, \boldsymbol{\theta})^\dagger. \quad (22)$$

The proof of this lemma is shown in Appendix B. Roughly, this lemma guarantees a good approximate eigen-decomposition of  $H$  for an optimized numerical solution of (6). This result can further support the following perturbation relation between the eigenspaces of  $H$  and  $\tilde{H}$ ,

$$\left\| P_{\lambda_k(H)} - P_{k, \tilde{H}} \right\|_F \ll 1. \quad (23)$$

Here  $P_{\lambda_k(H)}$  denotes the projector onto the  $k$ -th eigenspace of  $H$  and  $P_{k, \tilde{H}}$  is defined to be the projector of the eigenspace corresponding to the eigenvalues of  $\tilde{H}$  closest to  $\lambda_k$ , satisfying that  $\text{rank}(P_{k, \tilde{H}}) = \text{rank}(P_{\lambda_k(H)})$ . For each  $k$ , these projectors can be defined easily by listing the eigenvalues of  $H$  and  $\tilde{H}$  in increasing order. This can be useful for Hamiltonian simulations in low-energy subspace [30]. The following theorem shows that the perturbation relations hold uniformly for all  $k$ .

**Theorem 5.** *For any  $\epsilon' < \frac{\min_{\lambda_i \neq \lambda_j} |\lambda_i - \lambda_j|}{4 \max_k \text{rank}(P_{\lambda_k(H)})}$ , we assume a numerical solution  $(\mathbf{r}, \boldsymbol{\theta})$  of the cost function (6) as stated in Lemma 2 satisfying that*

$$\frac{F(\mathbf{r}, \boldsymbol{\theta})}{2^{n-1}} + 6(1 + \sqrt{F(\mathbf{r}, \boldsymbol{\theta})})\|H\|_F\sqrt{\epsilon} \leq \epsilon'. \quad (24)$$

*Then the  $k$ -th eigenspace of  $H$  is approximated by the set of eigenvectors of  $\tilde{H}$ , whose corresponding eigenvalues are closest to  $\lambda_k$ , namely,*

$$\left\| P_{k, \tilde{H}} - P_{\lambda_k(H)} \right\|_F \leq C \left( \frac{F(\mathbf{r}, \boldsymbol{\theta})}{2^{n-1}} + 6(1 + \sqrt{F(\mathbf{r}, \boldsymbol{\theta})})\|H\|_F\sqrt{\epsilon} \right), \quad (25)$$

*where the constant  $C > 0$  depends only on the spectrum of  $H$ .*

The proof is shown in Appendix B.

#### D. A randomized optimization algorithm

The cost of evaluating the gradient of the cost function (6) in Algorithm 1 scales as  $\mathcal{O}(d^4 M^2)$ . To be specific, the cardinality of set  $G_1$  scales as  $\mathcal{O}(d^2 M)$ , since  $K^\dagger H K$  in (3) consists of  $\mathcal{O}(d^2 M)$  Paulis. Then, to evaluate a partial derivative in (7), we execute a for-loop with  $\mathcal{O}(d^2 M)$  operations. Each operation computes the summand in (7) for a given  $P \in G_1$ , and the quantity  $\text{tr}(K^\dagger H K P)$  equals the coefficient of  $K^\dagger H K$  for the Pauli string  $P$ , which requires  $\mathcal{O}(dM)$  cost by looking for a pair of Pauli strings  $(P_1, P_2)$  such that for each  $Q_j$  in (1),  $P_1 Q_j P_2$  equals  $P$  up to a scaling factor. Since computing a partial derivative requires  $\mathcal{O}(d^3 M^2)$  and there are  $\mathcal{O}(d)$  partial derivatives, in total, we obtain the gradient of the term (3) in (6) with  $\mathcal{O}(d^4 M^2)$  operations. Similarly, we can deduce that  $\mathcal{O}(d^3)$  operations are required to compute the gradient of the term for orthogonality constraint in (6) by using the formulations in Lemma 1 and (A1). To summarize, the per-iteration cost of Algorithm 1 scales as  $\mathcal{O}(d^4 M^2)$ . To circumvent this quartic scaling of per-iteration cost, we propose a random coordinate version of Algorithm 1. This random coordinate optimization quadratically reduces the per-iteration cost from  $\mathcal{O}(d^4 M^2)$  to  $\mathcal{O}(d^2 M)$ , which is a significant improvement. To this end, we derive recursive relations for evaluating the cost function and its gradient. The algorithm is quite similar to Algorithm 1 except that a stochastic gradient estimate is introduced. Here

---

#### Algorithm 2: Random coordinate optimization algorithm

---

**Data:** initial guess  $\mathbf{x}_0 = (\mathbf{r}_0, \boldsymbol{\theta}_0)$  with  $\|\mathbf{r}_0\| = 1$ , learning rate schedule  $a_t \in (0, 1)$ , the maximal iteration number  $T$ , period  $p$

**Result:** approximate unitary  $K(\mathbf{r}_T, \boldsymbol{\theta}_T)$  for a KHK decomposition of  $H$

**for**  $t = 0 : T$  **do**

Sample indices  $\{i_t\} \subset [2^n]$ ;

Construct  $\mathbf{g}_t = \sum \frac{\partial F(\mathbf{x}_t)}{\partial i_t} \mathbf{e}_{i_t}$

$\mathbf{y}_t = \mathbf{x}_t - a_t \mathbf{g}_t$ ;

$\mathbf{x}_{t+1} = \left( \frac{\mathbf{r}(\mathbf{y}_t)}{\|\mathbf{r}(\mathbf{y}_t)\|}, \boldsymbol{\theta}(\mathbf{y}_t) \right)$ ;

**end**

---

we describe how to evaluate  $\mathbf{g}_t$  and the cost function in this algorithm.

Intuitively, if in each iteration, only a small fraction of the components in the parameters  $(\mathbf{r}, \boldsymbol{\theta})$  are updated, one expects that the change in the cost function (6) and its partial derivatives would be small. Roughly, in the first part of the cost function (6), we see that for sparse vectors  $\delta\mathbf{r}, \delta\boldsymbol{\theta}$ ,

$$\text{tr}(K(\mathbf{r} + \delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta})^\dagger H K(\mathbf{r} + \delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) P) = \text{tr}(K(\mathbf{r}, \boldsymbol{\theta})^\dagger H K(\mathbf{r}, \boldsymbol{\theta}) P) + \Delta, \quad (26)$$

where

$$\begin{aligned} \Delta &= 2 \text{tr}(K(\delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta})^\dagger H K(\mathbf{r} + \delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) P) + \text{tr}(K(\delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta})^\dagger H K(\delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) P) \\ &\quad + 2 \text{tr}(K(\mathbf{r}, \delta\boldsymbol{\theta})^\dagger H K(\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) P) + \text{tr}(K(\mathbf{r}, \delta\boldsymbol{\theta})^\dagger H K(\mathbf{r}, \delta\boldsymbol{\theta}) P). \end{aligned} \quad (27)$$

Notice that the four terms in the  $\Delta$  involve at least one of the sparse vectors as input. If we say that  $\delta\mathbf{r}$  and  $\delta\boldsymbol{\theta}$  are  $S$ -sparse vectors, then the cost of evaluating the four terms scales as  $\mathcal{O}(S)$ . The following specific formulations validate this argument.

$$K(\delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta})^\dagger H K(\mathbf{r} + \delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) = \left( \sum_{s=1}^S \delta r_{j_s} e^{i(\theta_{j_s} + \delta\theta_{j_s})} P_{j_s} \right)^\dagger H \left( \sum_{j=1}^d (r_j + \delta r_j) e^{i(\theta_j + \delta\theta_j)} P_j \right) \quad (28)$$

$$K(\delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta})^\dagger H K(\delta\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) = \left( \sum_{s=1}^S \delta r_{j_s} e^{i(\theta_{j_s} + \delta\theta_{j_s})} P_{j_s} \right)^\dagger H \left( \sum_{s=1}^S \delta r_{j_s} e^{i(\theta_{j_s} + \delta\theta_{j_s})} P_{j_s} \right) \quad (29)$$

$$K(\mathbf{r}, \delta\boldsymbol{\theta})^\dagger H K(\mathbf{r}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) = \left( \sum_{s=1}^S r_{j_s} (e^{i(\theta_{j_s} + \delta\theta_{j_s})} - e^{i\theta_{j_s}}) P_{j_s} \right)^\dagger H \left( \sum_{s=1}^S r_{j_s} e^{i(\theta_{j_s} + \delta\theta_{j_s})} P_{j_s} + \sum_{j \notin \{j_s\}_{s=1}^S} r_j e^{i\theta_j} P_j \right) \quad (30)$$

$$K(\mathbf{r}, \delta\boldsymbol{\theta})^\dagger H K(\mathbf{r}, \delta\boldsymbol{\theta}) = \left( \sum_{s=1}^S r_{j_s} (e^{i(\theta_{j_s} + \delta\theta_{j_s})} - e^{i\theta_{j_s}}) P_{j_s} \right)^\dagger H \left( \sum_{s=1}^S r_{j_s} (e^{i(\theta_{j_s} + \delta\theta_{j_s})} - e^{i\theta_{j_s}}) P_{j_s} \right) \quad (31)$$

Notice that in these formulas, there is at least one sum of  $S$  terms multiplied left or right to the Hamiltonian  $H$ . When these formulations are computed with the trace with the Pauli basis element  $P$  as in (26), only terms involving  $P$  are counted. That is, we iterate over the sum with  $S$  terms to find such terms, which takes only  $S$  steps with a for-loop, thereby yielding  $\mathcal{O}(S)$  computational cost. Together with these formulas, the relation (26) provides us with efficient recursive calculations for the first part of the cost function (6) and its gradient, as roughly summarized as

$$f(\mathbf{r}_{t+1}, \boldsymbol{\theta}_{t+1}) = f(\mathbf{r}_t, \boldsymbol{\theta}_t) + \tilde{\Delta}_t. \quad (32)$$

Here  $\tilde{\Delta}_t$  denotes the difference between  $f(\mathbf{r}_{t+1}, \boldsymbol{\theta}_{t+1})$  and  $f(\mathbf{r}_t, \boldsymbol{\theta}_t)$ , and is represented by the errors of the form  $\Delta$  in (27) and the definition (3), thereby depending on iteration step  $t$ . Estimating  $\tilde{\Delta}_t$  requires  $\mathcal{O}(d^2MS)$  computational cost, recalling that the first part of the cost function (6) consists of  $\mathcal{O}(d^2M)$  terms. Similarly, one can derive a recursive relation for the second part of the cost function (6) that involves the function  $\phi_P(\mathbf{r}, \boldsymbol{\theta})$ . The cost of evaluating the gradient of the cost function (6) is then  $\mathcal{O}(d^2MS^2)$ , since we construct a  $S$ -sparse stochastic gradient, each partial derivative is computed using (26), which takes  $\mathcal{O}(S)$ , and it is repeated  $\mathcal{O}(d^2M)$  times as the summation in (7) consists of  $\mathcal{O}(d^2M)$  terms. This cost dominates that of computing the gradient of the second part of the cost function (6). For more details, we refer the reader to our code, where recursive formulations are explicitly written.

With this randomization, the convergence analysis becomes non-trivial. Our algorithm performs the normalization on the radial parameters in  $\mathbf{r}_t$  at each iteration. If the full gradient is calculated as in Algorithm 1, a lower bound of the updated parameters,  $\mathbf{r}_{t+1}$ , can be derived analytically using Proposition 1 as in the proof of Theorem 4. However, in the case of randomized optimization, we cannot use Proposition 1, and thus deriving a lower bound of the norm of  $\mathbf{r}_{t+1}$  and a convergence guarantee is still an open issue. Nevertheless, with good initializations, the random coordinate algorithm Algorithm 2 performs well and outputs a good approximation of a diagonalization of the Hamiltonian, as numerically shown in Section V.

#### IV. A NEW FAMILY OF EFFICIENTLY DIAGONALIZABLE HAMILTONIANS

Typical Hamiltonians in quantum computing applications are represented by a linear combination of polynomially many Pauli strings. In the form (1) with  $M = \mathcal{O}(\text{poly}(n))$ , several families of Hamiltonians are found to be fast-forwarding or simulable on a quantum computer with polynomial

efforts, such as polynomially sized Lie algebras [16, 21, 27], mutually commuting Hamiltonians [2], and quantum diagonalizable Hamiltonians [26].

In this section, we identify families of Hamiltonians, and prove that their associated Lie algebras are exponentially large, but eigen-decompositions are polynomially small. To proceed, we follow the notion of quantum diagonalizable Hamiltonians [2, 8],

**Definition 1 (Quantum Diagonalizable Hamiltonian).** *A Hamiltonian is said to be quantum diagonalizable if*

$$H = UDU^\dagger, \quad (33)$$

where a unitary  $U$  and a diagonal  $D$  can be represented by polynomially sized circuits, respectively.

Within this family of Hamiltonians, we define a subset as follows,

**Definition 2.** *We consider a subset of the quantum diagonalizable Hamiltonians as follows,*

$$H = UDU^\dagger, \quad (34)$$

where a unitary  $U$  and a diagonal  $D$  can be represented by linear combinations of polynomially many Pauli strings, respectively.

Within the set of Hamiltonians in Definition 2, we show the existence of Hamiltonians whose Lie algebras are exponentially large. Specifically, the dimension of Lie algebra  $\mathfrak{g}(H)$  is  $4^n - 1$ , but the unitary of eigenvectors of the Hamiltonian  $H$ ,  $U$ , is represented by  $\mathcal{O}(\text{poly}(n))$  Pauli strings. The key idea is that the construction of the unitary of eigenvectors,  $U$ , involves  $2n - 1$  anticommuting Pauli strings, which is the maximum possible number for  $n$ -qubit systems [4].

**Example 1.** Consider a family of Hamiltonians defined as,

$$\begin{aligned} H &= UDU^\dagger, \\ U &= \left( \prod_{m=1}^{\lfloor \log n \rfloor} U_m \right) \exp(i\theta Z_2) \left( c_0 X_1 Y_2 + c_1 Z_1 Y_2 + c_2 Z_2 + \sum_{j=3}^n c_j X_2 Y_3 \dots Z_j \right), \\ D &= I + \sum_{j=1}^n d_j Y_j, \end{aligned} \quad (35)$$

where all parameters are non-zero, especially  $\theta \neq k\pi$  for integer  $k$ ,  $c_j$ 's are real and  $\sum_{j=0}^n c_j^2 = 1$ . Here, each  $U_m$  can be a Pauli rotation or the basic Clifford gate among  $S$ ,  $H$ , and  $CNOT$  and  $U_{\lfloor \log n \rfloor} \neq \exp(-i\theta Z_2)$ .

For any Hamiltonian in Example 1, we show that  $\mathfrak{g}(H) = \text{su}(2^n)$ , which yields that  $\dim(\mathfrak{g}(H)) = 4^n - 1$ . The proof is shown in Appendix C.

Using Example 1, one can also conceive similar examples with logarithmic repetitions as follows, which results in Hamiltonians whose Lie algebras are exponentially large,

**Example 2.** Appending unitaries of the form in Example 1,

$$U = \prod_{r=1}^{\lfloor \log n \rfloor} U_r, \quad (36)$$

reproduces Hamiltonians whose Lie algebras are exponentially large but polynomially small in Pauli strings,

$$H = UDU^\dagger, \quad (37)$$

where  $D$  is defined in Example 1.

Here we exploited the property of anticommuting Pauli strings [18] to conceive Hamiltonians whose Lie algebras are isomorphic to  $\text{su}(2^n)$  in the perspective of dynamical Lie algebra [31]. We could find another family of Hamiltonians by employing a complete pool of Pauli strings in the adaptive variational quantum eigensolver [34]. Since the set of Pauli strings there are used for preparing arbitrary real ground states, exponentially sized Lie algebras of Hamiltonians would be obtained. Together, one may be able to conceive a variety of Hamiltonians, whose eigen-decomposition is polynomially represented in Pauli strings as Definition 2, but the associated Lie algebra is exponentially large, following the idea of extensions of polynomially-sized Lie algebras in [27]. For such Hamiltonians, the per-iteration computational cost of Algorithm 1 scales polynomially, since the matrices  $K^\dagger HK$  and  $K^\dagger K$  involved in (6) are represented by polynomially many Paulis. By contrast, the Lie-diagonalization approaches [16, 21] would require an exponential scaling of per-iteration cost in this case, due to Lie algebra being exponentially large.

## V. NUMERICAL RESULT

In this section, we provide several numerical results to test the proposed algorithms. Our optimization algorithms rely on elementary calculations with Pauli strings, such as commutator calculations, conversion from Pauli strings to quaternary representations and vice versa. For these calculations, we used and extended a part of the code [21] in our optimization algorithms. First, we demonstrate Algorithm 1 for several Hamiltonians that are randomly generated in such a way that the unitary matrix  $K$  containing the eigenvectors and the diagonal matrix from the eigenvalues are sparse in terms of Pauli basis expansion.

Fig. 1 shows the optimization results for four Hamiltonians represented by different numbers of Pauli strings  $\text{len}(H) = 16, 24, 64, 96$ , where we denote  $\text{len}(H)$  to be the number of Pauli strings in the Hamiltonian  $H$ . To test Algorithm 1, we construct the Hamiltonians with the following steps (which may not belong to the families in Example 1 or Example 2): First, we sample a pair of polynomial numbers of Pauli strings, say,  $(\{P_{i_a}\}_{i_a \in I_a}, \{P_{i_b}\}_{i_b \in I_b})$ , where  $|I_a| = \mathcal{O}(\text{poly}(n))$  and  $|I_b| = \mathcal{O}(\text{poly}(n))$ . Here, each Pauli string  $P_{i_a}$  is diagonal. Second, we sample the same number of pairs of random real numbers  $(\{c_{i_a}\}_{i_a \in I_a}, \{c_{i_b}\}_{i_b \in I_b})$ . With these, a diagonal matrix is constructed as  $D = \sum_{i_a \in I_a} c_{i_a} P_{i_a}$ , and a unitary matrix is formulated as the product of Pauli rotations, say,  $U = \prod_{i_b \in I_b} \exp(ic_{i_b} P_{i_b})$ . Lastly, with this pair of unitary and diagonal, we construct a Hamiltonian that is represented by  $UDU^\dagger$ . For example, the Hamiltonian comprised of 16 Pauli strings is represented by a pair of numbers of Pauli strings (4, 2) for its eigen-decomposition. Similarly, for the cases where 24, 64, 96 Pauli strings are used to express Hamiltonians, the diagonals and the unitaries in their eigen-decompositions are represented by (4, 4), (6, 2), (6, 4) pairs of numbers of Pauli strings, respectively. That is, for the cases  $\text{len}(H) = 16, 24, 64, 96$ , the number of parameters in  $K$  in (3) is 8, 32, 8, 32, respectively.

We observe that in all cases, the algorithm Algorithm 1 converges rather quickly. To study the convergence in response to Theorem 4, we monitored the value of  $\alpha$  in Assumption 1 during the iteration. On average, the values of  $\alpha$  for the cases where  $\text{len}(H) = 16, 24, 64$  remain close to 1. In these cases, the optimization results are shown to converge almost linearly as in Fig. 1,

$\Delta$	Initial error	Final error
0.1	3.870	0.028
0.5	2.073	0.010
0.8	0.800	0.003
1.2	0.765	0.003

(a) XXZ model

$U$	Initial error	Final error
2	2.529	0.042
4	1.000	0.020
5	0.441	0.008
7	0.350	0.007

(b) Hubbard model

TABLE I: Initial and final errors,  $\|H - \tilde{H}\|_F$ , for the optimization results in Fig. 2

which is consistent with the theoretical result in Theorem 4. On the other hand, in the case where  $\text{len}(H) = 96$ , the value of  $\alpha$  seems to deviate from 1 and the convergence becomes sublinear as illustrated in Fig. 1, again consistent with Theorem 4.

To further illustrate the optimization properties of our algorithm, we study four-qubit examples, including an XXZ model and a Hubbard model, and the results are shown in Fig. 2. In the numerical tests, four different initializations are set by changing the parameters in the models. Specifically, the XXZ model is given by

$$H_{XXZ} = J \sum_{i=1}^{n-1} (X_i X_{i+1} + Y_i Y_{i+1}) + \Delta \sum_{i=1}^{n-1} Z_i Z_{i+1}, \quad (38)$$

and for the Hubbard model, the Hamiltonian is expressed as

$$H = -\frac{t}{2} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{j=1}^{n-1} (X_{j+1,\sigma} X_{j,\sigma} + Y_{j+1,\sigma} Y_{j,\sigma}) + \frac{U}{4} \sum_{j=1}^n (-Z_{j,\uparrow} - Z_{j,\downarrow} + Z_{j,\uparrow} Z_{j,\downarrow}). \quad (39)$$

Here we omit the identity matrix, which does not affect diagonalization.

For the XXZ model, we aim to diagonalize the Hamiltonian with parameters  $J = 1$ . For the Hubbard model, we set  $t = 1$ . Meanwhile, as shown in Table I, we choose several different values for  $\Delta$  or  $U$  and run the optimization algorithms. Specifically, with a Hamiltonian defined by a different value of  $\Delta$ , for instance, we compute its eigen-decomposition using an eigensolver, obtain a unitary  $K$ , expand  $K$  in terms of Pauli strings as (3) and use the coefficients as the initial parameters for the optimization (6) with the Hamiltonian defined by the target value of  $\Delta$ .

In these experiments, we used Algorithm 2 since the Pauli expansion of the unitary of eigenvectors and the diagonal of eigenvalues are not sparse in these cases, and Algorithm 1 may not be efficiently implementable. As shown in Fig. 2, the numerical results suggest the fast convergence of Algorithm 2 to diagonalizations of the target Hamiltonians, especially when the initial parameter is well selected. These findings suggest that the optimized parameter set obtained from (6) for one Hamiltonian is an excellent initialization for our optimization procedure when diagonalizing another Hamiltonian with slightly perturbed physical parameters.

## VI. DISCUSSION

In the previous sections, we identified a family of fast-forwarding Hamiltonians (Definition 2) and developed an efficient classical algorithm (Algorithm 1). In this section, we examine the significance of our results from various perspectives and discuss related work.

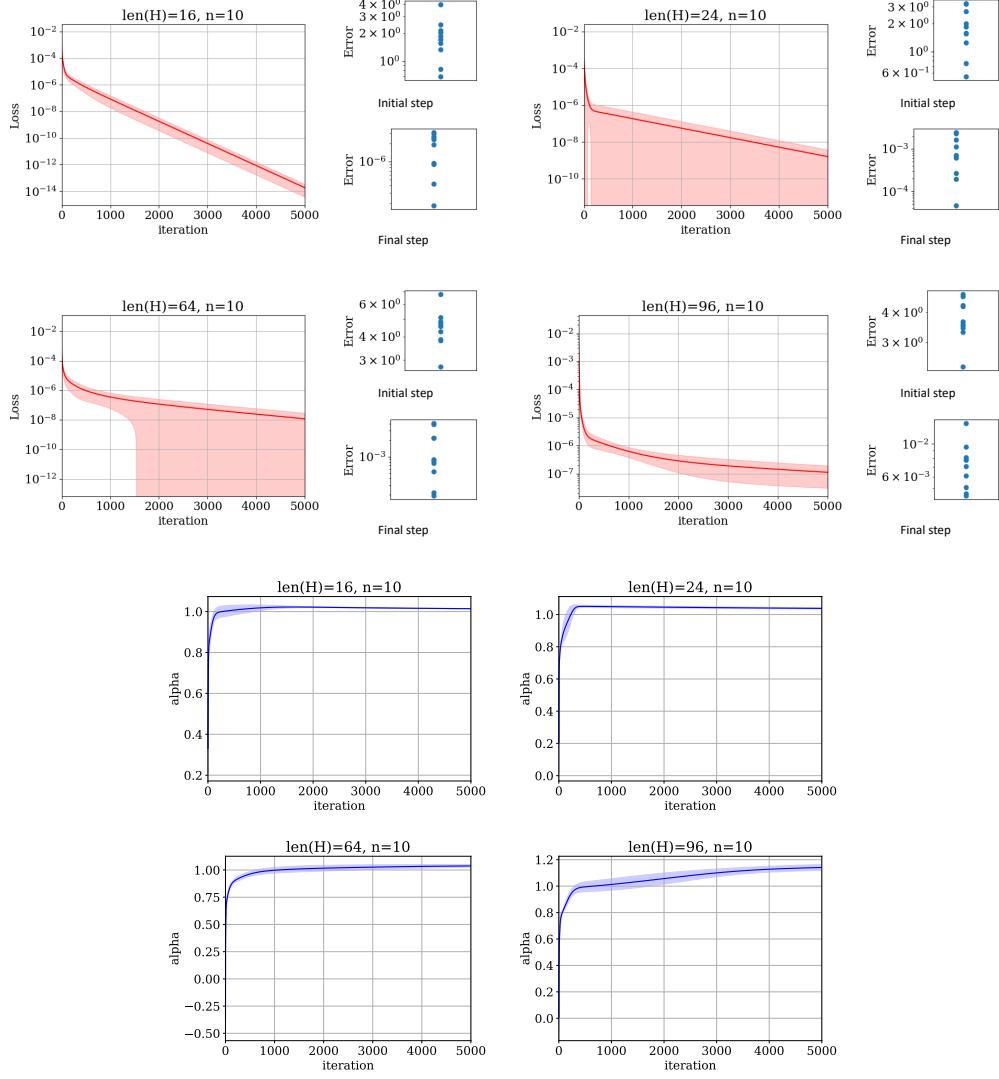


FIG. 1: Optimization results (Top) and the corresponding values of  $\alpha$  with  $\mu = 1$  in the KL condition (1) (Bottom) for four different Hamiltonians randomly generated comprised of 16, 24, 64 and 96 Pauli strings for  $n$ -qubit system ( $n = 10$ ). The error in the optimization results indicates the Frobenius error,  $\|H - \tilde{H}\|_F$ , where  $H$  and  $\tilde{H}$  denote the original Hamiltonian and an approximation defined in Lemma 2. We compare the errors estimated at the first step and the last step of the optimization.

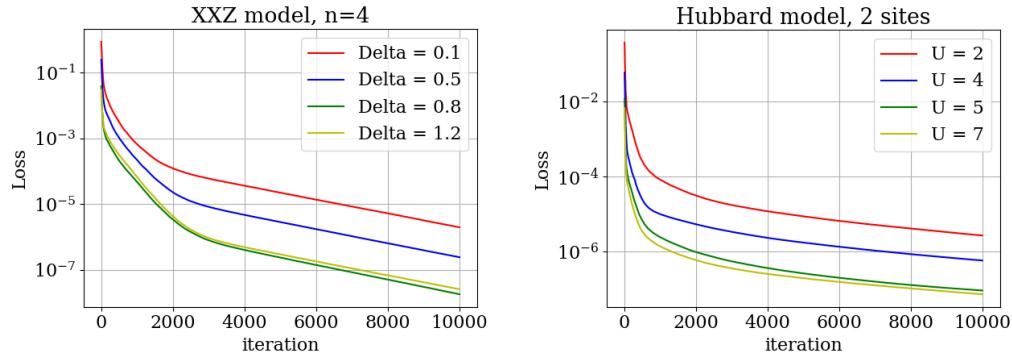


FIG. 2: Optimization results for XXZ model and Hubbard model with four different choices of parameters  $\Delta$  or  $U$  in Eqs. (38) and (39) as listed in Table I, which give four different initializations.

The previous works [8, 26] have raised an important open question: is the set of quantum diagonalizable Hamiltonians (Definition 1) smaller than that of fast-forwarding Hamiltonians? Addressing this question may first require identifying the Hamiltonians that lie within the family of quantum diagonalizable Hamiltonians (Definition 1). To date, much of the existing literature has focused on Hamiltonians associated with polynomially sized Lie algebras (PLA), such as quadratic fermionic and certain bosonic Hamiltonians [16], non-contextual Pauli and fermionic mean-field Hamiltonians [27], and other further examples thanks to the existence of a polynomially sized ansatz with Pauli rotations [21]. In contrast, the case of Hamiltonians generated by exponentially large Lie algebras has remained largely unexplored in this context, despite the fact that such Hamiltonians encompass a wide range of physically and computationally significant models [39]. This raises an important question: Is every quantum diagonalizable Hamiltonian necessarily associated with a PLA? From the examples presented in Example 1 and Example 2, we provide evidence that the class of quantum-diagonalizable Hamiltonians *strictly* contains the PLA Hamiltonians. As illustrated in Fig. 3, this result establishes a previously unrecognized relationship between these two classes. Namely, the former is strictly larger than the latter. In other words, there remains an opportunity to construct efficient diagonalization circuits for certain fast-forwardable Hamiltonians whose underlying Lie algebras are exponentially large, beyond the scope of conventional PLA-based approaches.

Second, we introduced a classical optimization algorithm (Algorithm 1) tailored to the family of Hamiltonians in Definition 2. While, in principle, existing quantum algorithms such as the variational fast-forwarding (VFF) method [8] and the double-bracket diagonalization algorithm [14] are applicable to this family, several limitations hinder their practical use. First, the optimization landscape of VFF may involve local minima and saddle points as other variational quantum algorithms. Although strategies such as adaptive ansatz growth (e.g., adaptive VQE) have been proposed to mitigate these issues, they do not offer guarantees against convergence to suboptimal points in general, and may lead to over-parameterization in practice [11, 15]. Furthermore, implementing the coherent Hamiltonian simulations, which is required for the training, introduces additional overhead and complexity. By contrast, our algorithm optimizes the cost function that is analytically guaranteed to have only global minima as stationary points according to Theorem 3. Specifically, any non-zero global minimum directly yields a valid diagonalization unitary, and our algorithm converges to only

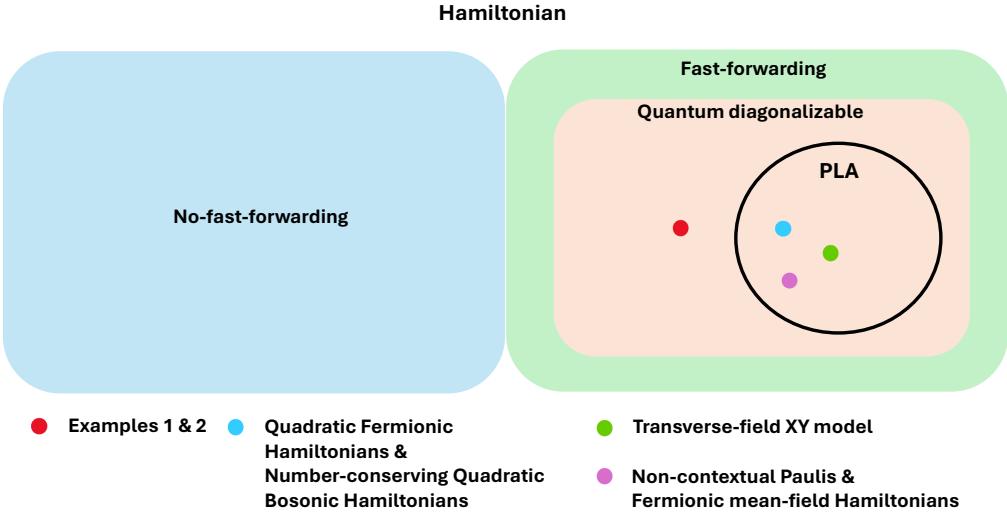


FIG. 3: The diagram illustrates examples of quantum diagonalizable Hamiltonians: this work (red), [16] (blue), [21] (green), and [27] (purple).

a non-zero minimum due to the normalization step enforced at each iteration. As a result, our approach circumvents the suboptimality issues that are inherent in VFF, offering a more direct and reliable route to diagonalization, without any quantum resource. Another approach by Gluza [14] implements a double-bracket dynamics to construct a diagonalizing unitary. This technique is constructive and broadly applicable, due to its non-increasing property of the off-diagonal terms. However, the family of Hamiltonians for which the algorithm operates efficiently remains unclear from a complexity-theory perspective. Moreover, the required circuit depth increases exponentially with the number of iterations, posing serious challenges to scalability in practical implementations. Therefore, we think that the proposed classical diagonalization algorithm is more suitable than the two quantum algorithms for the family of Hamiltonians considered. This conclusion aligns with insights from the literature of dequantization algorithms, where certain classical algorithms, under additional assumptions, are shown to perform comparably to their quantum counterparts [12, 33], and thus reshape expectations around quantum advantage in specific algorithmic contexts.

We remark that the formulation (6) is equivalent or similar to previous Lie-diagonalization approaches. For example, Somma's Lie-diagonalization algorithm [16, 32] is viewed as a variant of Jacobi's eigenvalue algorithm, iteratively minimizing off-diagonal terms, which is conceptually aligned with Algorithm 1. However, their algorithm is more suitable for PLA Hamiltonians, whereas our algorithm is designed to handle a family of Hamiltonians with exponentially large Lie algebras (e.g. Definition 2). On the other hand, the algorithm in [21] finds a Cartan decomposition of Hamiltonian, which has been widely studied in the context of quantum control theory [9, 10, 19]. Their algorithm assumes prior algebraic knowledge of the Cartan decomposition of a given Hamiltonian and essentially seeks to identify a parameterized unitary  $K$  such that

$$\begin{aligned} [K^\dagger H K, v] &= 0, \quad \text{for some } v \in \mathfrak{h}, \\ K^\dagger H K &\in \mathfrak{m}, \end{aligned} \tag{40}$$

as deduced from the proof of [21, Theorem.5, Appendix D]. Here, a Pauli string  $v$  satisfies that its corresponding exponential  $e^{itv}$  for real  $t$  is dense in the Lie group,  $e^{i\mathfrak{h}}$ , associated to a Cartan subalgebra  $\mathfrak{h}$ . However, the condition in Eq. (40) can be succinctly expressed as

$$\text{tr}(K^\dagger H K P) = 0, \quad (41)$$

for any Pauli string  $P \notin \mathfrak{h}$  with a unitary  $K$ . Notice that this condition exactly corresponds to a non-zero global minimum of the cost function (6). Therefore, given the same algebraic knowledge of the Cartan decomposition, our formulation in (6) can be equivalently interpreted as the optimization problem presented in [21], by expressing  $K$  and  $K^\dagger H K$  within the framework of Cartan decomposition. Nevertheless, we see that these Lie-diagonalization algorithms may be inefficient in general for the family of Hamiltonians in Definition 2 due to the existence of exponentially-large Lie algebras in Example 1 and Example 2.

## VII. CONCLUSION

In this work, we present classical optimization algorithms for Hamiltonian diagonalization and identify a family of Hamiltonians for which these algorithms remain efficient, while existing algorithms are not. Specifically, the proposed algorithms exhibit only polynomial per-iteration scaling for Hamiltonians whose eigen-decompositions consist of a diagonal matrix and a unitary operator that are both sparse in the Pauli basis, and whose optimization landscapes admit only global minima as stationary points. To this end, we formulate a cost function in terms of Pauli strings, which constitutes the off-diagonal terms of the Hamiltonian with an orthogonality constraint. We establish that any non-zero stationary point of the cost function corresponds to a global minimum, and eventually a diagonalization of the Hamiltonian. To find such a non-zero stationary point, we propose a simple optimization algorithm such that the gradient descent is implemented at each iteration with normalization. We show that this algorithm converges under a reasonable non-convex condition that is validated in numerical experiments. Our analysis also shows the role of a good initialization in enabling fast convergence.

Toward practical implementations, we also introduced a randomized-coordinate variant of our optimization algorithm that delivers a more efficient iteration scheme than its deterministic counterpart. By formulating a tailored update rule, this randomized approach achieves a quadratic scaling per-iteration cost with respect to the Hilbert-space dimension. Another key numerical observation is that a continuation strategy can accelerate convergence: the optimal parameters obtained for one Hamiltonian serve as an excellent initialization when diagonalizing a nearby Hamiltonian whose physical parameters differ only slightly. Consequently, our algorithms are well-suited for successively diagonalizing families of Hamiltonians that vary smoothly in a control parameter and for applications where rapid, “fast-forward” Hamiltonian simulation is crucial. A notable example is Hamiltonian simulation in the interaction picture [25], as well as certain ODE systems [1].

## CODE AVAILABILITY

Code used for the current study are available at the following GitHub repository:  
<https://github.com/TKmath/Classical-algorithms-for-diagonalizing-quantum-Hamiltonians>

## ACKNOWLEDGEMENT

This research was supported by Quantum Simulator Development Project for Materials Innovation through the National Research Foundation of Korea (NRF) funded by the Korean government (Ministry of Science and ICT(MSIT))(RS-2023-NR119931). We used resources of the Center for Advanced Computation at Korea Institute for Advanced Study and the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No.DE-AC02-05CH11231. SC was supported by a KIAS Individual Grant (CG090601) at Korea Institute for Advanced Study. XL is supported by the NSF Grant DMS-2411120. TK is supported by a KIAS Individual Grant (CG096001) at Korea Institute for Advanced Study.

## CONFLICTS OF INTERESTS

All authors declare no financial or non-financial competing interest

## AUTHOR CONTRIBUTIONS

TK, SC, and XL conceptualized the project. Analyses and numerical experiments were led by TK and discussed with SC, HP and XL. The manuscript was written by TK, and XL.

## **Appendix A: Optimization with only global minima and convergence analysis**

In this section, we present proofs for Theorem 3 and Theorem 4. First, we prove Theorem 3.

*Proof.* Let  $(\mathbf{r}_c, \boldsymbol{\theta}_c)$  be a non-zero stationary point, that is  $\mathbf{r}_c \neq 0$ . Denote  $K_c = K(\mathbf{r}_c, \boldsymbol{\theta}_c)$ . By (6) and (7), it follows at the stationary point,

$$\begin{aligned}
 0 &= \frac{\partial F}{\partial r_j} \\
 &= \frac{\partial f}{\partial r_j} + 2 \sum_{P \in G_2} \phi_P(\mathbf{r}_c, \boldsymbol{\theta}_c) \frac{\phi_P(\mathbf{r}_c, \boldsymbol{\theta}_c)}{\partial r_j} \\
 &= \frac{\partial f}{\partial r_j} + 4 \sum_{P \in G_2} \phi_P(\mathbf{r}_c, \boldsymbol{\theta}_c) c_{j,P} r_{j_P} \exp(i(\theta_j - \theta_{j_P})). \tag{A1}
 \end{aligned}$$

Similar to the proof of Theorem 2, by multiplying  $r_j$  to (A1) and summing over  $j$ 's, we arrive at

$$\begin{aligned} 0 &= 4f(\mathbf{r}_c, \boldsymbol{\theta}_c) + 4 \sum_j r_j \left( \sum_{P \in G_2} \phi_P(\mathbf{r}_c, \boldsymbol{\theta}_c) c_{j,P} r_{j_P} \exp(i(\theta_j - \theta_{j_P})) \right) \\ &= 4f(\mathbf{r}_c, \boldsymbol{\theta}_c) + 4 \sum_{P \in G_2} \phi_P(\mathbf{r}_c, \boldsymbol{\theta}_c) \left( \sum_j c_{j,P} r_j r_{j_P} \exp(i(\theta_j - \theta_{j_P})) \right) \\ &= 4f(\mathbf{r}_c, \boldsymbol{\theta}_c) + 4 \sum_{P \in G_2} \phi_P(\mathbf{r}_c, \boldsymbol{\theta}_c)^2 = 4F(\mathbf{r}_c, \boldsymbol{\theta}_c), \end{aligned} \quad (\text{A2})$$

by definition of  $\phi_P$  in Lemma 1. Since  $F(\mathbf{r}_c, \boldsymbol{\theta}_c) = 0$ ,  $(\mathbf{r}_c, \boldsymbol{\theta}_c)$  is a global minimum of the cost function in (6).

Now we prove that  $K$  is unitary up to scaling. Noting that  $\sum_{P \in G_2} \phi_P(\mathbf{r}_c, \boldsymbol{\theta}_c)^2 = 0$  from (A2), we know from Lemma 1 that

$$K(\mathbf{r}_c, \boldsymbol{\theta}_c)^\dagger K(\mathbf{r}_c, \boldsymbol{\theta}_c) = \|\mathbf{r}_c\|^2 I. \quad (\text{A3})$$

Additionally, since  $f(\mathbf{r}_c, \boldsymbol{\theta}_c) = 0$ , it immediately follows that  $K(\mathbf{r}_c, \boldsymbol{\theta}_c)^\dagger H K(\mathbf{r}_c, \boldsymbol{\theta}_c)$  is a diagonal matrix, which results that

$$H = \frac{1}{\|\mathbf{r}\|^4} K(\mathbf{r}_c, \boldsymbol{\theta}_c) h(\mathbf{r}_c, \boldsymbol{\theta}_c) K(\mathbf{r}_c, \boldsymbol{\theta}_c)^\dagger, \quad (\text{A4})$$

where  $h(\mathbf{r}_c, \boldsymbol{\theta}_c) := K(\mathbf{r}_c, \boldsymbol{\theta}_c)^\dagger H K(\mathbf{r}_c, \boldsymbol{\theta}_c)$ . This completes the proof.  $\square$

This proof shows that the optimization problem we consider has only global minima as stationary points, and especially any non-zero stationary point yields a diagonalization unitary  $K$ . Moreover, the following proof guarantees that Algorithm 1 converges to a non-zero stationary point.

*Proof.* Since the function in (6) is a polynomial of degree 4 with respect to  $\mathbf{r}$  and  $\boldsymbol{\theta}$  appear there as the phase factors that do not affect the magnitude of the function, we can find a constant  $L > 0$  such that

$$\max_{\|\mathbf{r}(\mathbf{x})\| \leq 2} \|\nabla^2 F(\mathbf{x})\| \leq L, \quad (\text{A5})$$

in the neighborhood of radius 2 defined in the  $\mathbf{r}$  direction. To proceed, we derive a recursive inequality for Algorithm 1. Suppose that  $\|\mathbf{r}(\mathbf{x}_t)\| = 1$ , as Algorithm 1 calls for normalization about  $\mathbf{r}$  at each iteration.

From the definition of  $\mathbf{y}_t$  in Algorithm 1 and the condition on  $\mathbf{x}_t$ , we notice that

$$\begin{aligned} \|\mathbf{r}(\mathbf{y}_t)\|^2 &= \sum_j ((\mathbf{r}(\mathbf{x}_t))_j - a \frac{\partial F}{\partial r_j}(\mathbf{x}_t))^2 = \|\mathbf{r}(\mathbf{x}_t)\|^2 - 2a \mathbf{r}(\mathbf{x}_t) \cdot \nabla_r F(\mathbf{x}_t) + a^2 \|\nabla_r F(\mathbf{x}_t)\|^2 \\ &= 1 - 8aF(\mathbf{x}_t) + a^2 \|\nabla_r F(\mathbf{x}_t)\|^2 \end{aligned} \quad (\text{A6})$$

where  $\nabla_r F := (\frac{\partial F}{\partial r_1}, \dots, \frac{\partial F}{\partial r_N})^T$ . The last equality follows from Proposition 1. We derive a lower bound for  $\|\mathbf{r}(\mathbf{y}_t)\|$ . By applying the Cauchy-Schwarz inequality to Proposition 1 in (A6), we have

$$\|\mathbf{r}(\mathbf{y}_t)\|^2 \geq 1 - 8aF(\mathbf{x}_t) + 16a^2 F(\mathbf{x}_t)^2 \geq 1 - 8aF(\mathbf{x}_t). \quad (\text{A7})$$

Then for any  $a \leq \frac{2}{L}$ , we achieve that

$$\begin{aligned} F(\mathbf{y}_t) &\leq F(\mathbf{x}_t) - a\|\nabla F(\mathbf{x}_t)\|^2 + \frac{La^2}{2}\|\nabla F(\mathbf{x}_t)\|^2 \\ &\leq F(\mathbf{x}_t) - \frac{a}{2}\|\nabla F(\mathbf{x}_t)\|^2 \\ &\leq (1 - 2\mu a F(\mathbf{x}_t)^{\alpha-1})F(\mathbf{x}_t). \end{aligned} \tag{A8}$$

The last inequality holds by the assumption (1). By (A7) and (A8), we observe that

$$\begin{aligned} F(\mathbf{x}_{t+1}) &= \frac{1}{\|\mathbf{r}(\mathbf{y}_t)\|^4}F(\mathbf{y}_t) \leq \frac{1 - 2\mu a F(\mathbf{x}_t)^{\alpha-1}}{(1 - 8a F(\mathbf{x}_t))^2}F(\mathbf{x}_t) \\ &\leq \frac{1 - 2\mu a F(\mathbf{x}_t)^{\alpha-1}}{1 - 16a F(\mathbf{x}_t)}F(\mathbf{x}_t) \\ &\leq (1 - (2\mu - 16F(\mathbf{x}_t)^{2-\alpha})a F(\mathbf{x}_t)^{\alpha-1})F(\mathbf{x}_t) \\ &\leq (1 - \mu a F(\mathbf{x}_t)^{\alpha-1})F(\mathbf{x}_t), \end{aligned} \tag{A9}$$

when  $F(\mathbf{x}_t) \leq (\frac{\mu}{16})^{\frac{1}{2-\alpha}}$ . To find the number of iterations  $T$  to achieve  $\epsilon$  precision, we solve the following inequality,

$$\epsilon \leq (1 - \mu a \epsilon^{\alpha-1})^T F(\mathbf{x}_0). \tag{A10}$$

Therefore, for a reasonable initial guess  $\mathbf{x}_0$ , Algorithm 1 requires the number of iterations scaling as

$$T = \mathcal{O}\left(\frac{L}{\mu \epsilon^{\alpha-1}} \log \frac{F(\mathbf{x}_0)}{\epsilon}\right). \tag{A11}$$

□

## Appendix B: Sensitivity analysis for numerical solutions

In this section, we prove Lemma 2 and Theorem 5. First, we prove Lemma 2.

*Proof.* For ease of notation in analysis, we omit the parameters  $\mathbf{r}, \boldsymbol{\theta}$  unless stated otherwise. Recalling the set  $G_1$  in (3) and denoting  $G_3 = \{\otimes_{i=1}^n \sigma^{(i)} : \sigma^{(i)} \in \{I, X, Y, Z\} \text{ and } \forall i \in [n], \sigma^{(i)} \in \{I, Z\}\}$ , we define

$$h_0 = \frac{1}{2^n} \sum_{P \in G_3} \text{tr}(K^\dagger H K P) P, \quad \Delta = \frac{1}{2^n} \sum_{P \in G_1} \text{tr}(K^\dagger H K P) P, \tag{B1}$$

which sum to the  $K^\dagger H K$ , thus verifying Eq. (21). By definition (3), we observe that

$$\|\Delta\|_F^2 = \frac{1}{2^n} \sum_{P \in G_1} \text{tr}(K^\dagger H K P)^2 = \frac{f}{2^n} \leq \frac{F}{2^n}, \tag{B2}$$

since  $f \leq F$  by (6). This proves the first statement of Lemma 2. For the second statement, by our assumptions, we have from the proof of Lemma 1 that,

$$\|K^\dagger K - I\|_F^2 = \left\| \sum_{P \in G_2} \phi_P(\mathbf{r}, \boldsymbol{\theta}) P \right\|_F^2 = 2^n \sum_{P \in G_2} \phi_P(\mathbf{r}, \boldsymbol{\theta})^2 \leq \epsilon \leq \frac{1}{4}. \quad (\text{B3})$$

Note that the inverse of  $K$  exists, and

$$\|K^{-1}\|_2^2 = \|(K^\dagger K)^{-1}\|_2 = \|(K^\dagger K - I + I)^{-1}\|_2 \leq \frac{1}{1 - \sqrt{\epsilon}} \leq 2. \quad (\text{B4})$$

From this, we have the error in the spectral norm,

$$\begin{aligned} \|H - \tilde{H}\|_2 &= \left\| (K^\dagger)^{-1} K^\dagger (H - \tilde{H}) K K^{-1} \right\|_2 \\ &= \left\| (K^\dagger)^{-1} (h_0 + \Delta - K^\dagger \tilde{H} K) K^{-1} \right\|_2 \\ &= \left\| (K^\dagger)^{-1} (h_0 + \Delta - K^\dagger K h_0 K^\dagger K) K^{-1} \right\|_2 \\ &\leq 2 \|h_0 + \Delta - K^\dagger K h_0 K^\dagger K\|_2. \end{aligned} \quad (\text{B5})$$

The term in the last line is bounded above as

$$\begin{aligned} &\|h_0 + \Delta - K^\dagger K h_0 K^\dagger K\|_F \\ &= \|h_0 + \Delta - (K^\dagger K - I + I) h_0 (K^\dagger K - I + I)\|_F \\ &= \|\Delta - (K^\dagger K - I) h_0 (K^\dagger K - I) - (K^\dagger K - I) h_0 - h_0 (K^\dagger K - I)\|_F \\ &\leq \frac{F}{2^n} + 3 \|h_0\|_F \sqrt{\epsilon}. \end{aligned} \quad (\text{B6})$$

Finally, noting that

$$\|h_0\|_F^2 = \frac{1}{2^n} \sum_{p \in \mathfrak{h}} \text{tr}(K^\dagger H K p)^2 \leq \frac{1}{2^n} \|K^\dagger H K\|_F^2 \leq \frac{1}{2^n} \|K^\dagger K\|_F^2 \|H\|_F^2 \leq (1 + \sqrt{F})^2 \|H\|_F^2, \quad (\text{B7})$$

with the above results, we obtain

$$\|H - \tilde{H}\|_2 \leq \frac{F}{2^{n-1}} + 6(1 + \sqrt{F}) \|H\|_F \sqrt{\epsilon}, \quad (\text{B8})$$

which proves the second statement of Lemma 2.  $\square$

This completes the proof of Lemma 2. Using the result, we prove Theorem 5 that gives an approximation error bound of diagonalization of Hamiltonian for a numerical solution obtained from our algorithm.

*Proof.* From Lemma 2, we notice that  $\|H - \tilde{H}\|_2 \leq \epsilon'$ . Next, we refer to the proof of [20, Lemma 5] from which we can deduce that

$$\langle \tilde{\psi}_j | P_{\lambda_k(H)} | \tilde{\psi}_j \rangle \geq 1 - (C + 1)\epsilon', \quad (\text{B9})$$

where  $\tilde{\psi}_j$  denotes any eigenvector of  $P_{k,\tilde{H}}$  and the constant  $C$  depends on the eigenvalues of  $H$  by considering a fixed polynomial approximation in the Stone-Weierstrass theorem to the indicator function in [20, eq (38)]. By summing this inequality over  $j$ 's, we have

$$\text{tr} \left( P_{k,\tilde{H}} P_{\lambda_k(H)} \right) \geq \text{rank}(P_{\lambda_k(H)}) (1 - (C + 1)\epsilon'), \quad (\text{B10})$$

since  $\text{rank}(P_{k,\tilde{H}}) = \text{rank}(P_{\lambda_k(H)})$ , and therefore

$$\begin{aligned} \left\| P_{k,\tilde{H}} - P_{\lambda_k(H)} \right\|_F^2 &\leq 2\text{rank}(P_{\lambda_k(H)}) - 2\text{tr} \left( P_{k,\tilde{H}} P_{\lambda_k(H)} \right) \\ &\leq 2\text{rank}(P_{\lambda_k(H)})(C + 1)\epsilon' \leq 2 \max_k \text{rank}(P_{\lambda_k(H)})(C + 1)\epsilon'. \end{aligned} \quad (\text{B11})$$

Therefore, the statement is proven.  $\square$

### Appendix C: Quantum diagonalizable Hamiltonians whose Lie algebras are exponentially large

In this section, we prove that the Lie algebras of Hamiltonians in Example 1 are exponentially sized.

*Proof.* We consider the trivial case where  $\prod_{m=1}^{\lfloor \log n \rfloor} U_m = I$ . If  $\mathfrak{g}(H) = \text{su}(2^n)$  in this case, then any Hamiltonian in Example 1 would satisfy the same property.

To proceed, we make use of the result in [31], which states that the set,

$$\{Z_1, Z_2, X_1, X_2, Z_1Z_2\} \cup \{X_2Y_3\dots Z_j, Z_2Y_3\dots X_j | 3 \leq j \leq n\}, \quad (\text{C1})$$

generates all Pauli strings in the sense of dynamical Lie algebra. Noticing that the second factor of  $U$  consists of mutually-anticommuting Pauli strings, we see that the constraint that  $\sum_{j=0}^n c_j^2 = 1$  and  $c_j$ 's are real suffices to ensure that  $U$  is a unitary, a property of the anticommuting Pauli strings [18, 29]. Now we prove the statement by showing that all Pauli strings in the set (C1) are contained in the Hamiltonian  $H$  and its dynamical Lie algebra.

First, we observe that  $U$  contains the identity  $I$  as its component, since  $i \sin \theta Z_2 \times c_2 Z_2 = ic_2 \sin \theta I$  and  $c_2, \theta \neq 0$ . This implies that  $H$  contains the set  $\{X_2Y_3\dots Z_j | 3 \leq j \leq n\}$ , since the elements of Pauli string are chosen from  $U \times I \times ic_2 \sin \theta I$  in the multiplication  $UDU^\dagger$ . Similarly, we can find that  $Z_1Y_2 \times Y_2 \times I = Z_1$ ,  $X_1Y_2 \times Y_2 \times I = X_1$ ,  $Z_2 \times I \times I = Z_2$ ,  $Z_2 \times Y_2 \times I = X_2$ , and  $Z_1X_2 \times Y_2 \times I = Z_1Z_2$ . Notice that  $Z_1X_2$  is in  $U$  due to the multiplication,  $i \sin \theta Z_2 \times c_1 Z_1 Y_2$ . Lastly, we are left to show that the Pauli strings in  $\{Z_2Y_3\dots X_j | 3 \leq j \leq n\}$  are contained in  $\mathfrak{g}(H)$ . We observe that since  $I \times Y_j \times I = Y_j$ ,  $Y_j$ 's are in  $H$ . Since  $H$  contains the set  $\{X_2Y_3\dots Z_j | 3 \leq j \leq n\}$  as shown above, and  $[[X_2Y_3\dots Z_j, Y_2], Y_j]$  equals  $Z_2Y_3\dots X_j$  up to a scaling factor, it happens that  $\mathfrak{g}(H)$  also contains the set  $\{Z_2Y_3\dots X_j | 3 \leq j \leq n\}$ . Therefore,  $\mathfrak{g}(H)$  contains the generating set (C1) and all  $n$ -qubit Pauli strings.  $\square$

---

[1] Dong An, Jin-Peng Liu, Daochen Wang, and Qi Zhao. A theory of quantum differential equation solvers: limitations and fast-forwarding. *arXiv preprint arXiv:2211.05246*, 2022.

- [2] Yosi Atia and Dorit Aharonov. Fast-forwarding of hamiltonians and exponentially precise measurements. *Nature communications*, 8(1):1572, 2017.
- [3] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Physical review letters*, 114(9):090502, 2015.
- [4] Xavier Bonet-Monroig, Ryan Babbush, and Thomas E O’Brien. Nearly optimal measurement scheduling for partial tomography of quantum states. *Physical Review X*, 10(3):031064, 2020.
- [5] Andrew M Childs and Yuan Su. Nearly optimal lattice simulation by product formulas. *Physical review letters*, 123(5):050503, 2019.
- [6] Andrew M Childs, Yuan Su, Minh C Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Physical Review X*, 11(1):011020, 2021.
- [7] Moody T. Chu. Lax dynamics for cartan decomposition with applications to hamiltonian simulation. *IMA Journal of Numerical Analysis*, 44(3):1406–1434, 2024.
- [8] Cristina Cirstoiu, Zoe Holmes, Joseph Iosue, Lukasz Cincio, Patrick J Coles, and Andrew Sornborger. Variational fast forwarding for quantum simulation beyond the coherence time. *npj Quantum Information*, 6(1):82, 2020.
- [9] Domenico D’Alessandro and Francesca Albertini. Quantum symmetries and cartan decompositions in arbitrary dimensions. *Journal of Physics A: Mathematical and Theoretical*, 40(10):2439, 2007.
- [10] Henrique N Sá Earp and Jiannis K Pachos. A constructive algorithm for the cartan decomposition of su (2n). *Journal of mathematical physics*, 46(8), 2005.
- [11] César Feniou, Muhammad Hassan, Diata Traoré, Emmanuel Giner, Yvon Maday, and Jean-Philip Piquemal. Overlap-adapt-vqe: practical quantum chemistry on quantum computers via overlap-guided compact ansätze. *Communications Physics*, 6(1):192, 2023.
- [12] Sevag Gharibian and François Le Gall. Dequantizing the quantum singular value transformation: hardness and applications to quantum chemistry and the quantum pcp conjecture. In *Proceedings of the 54th annual ACM SIGACT symposium on theory of computing*, pages 19–32, 2022.
- [13] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, pages 193–204, 2019.
- [14] Marek Gluza. Double-bracket quantum algorithms for diagonalization. *Quantum*, 8:1316, 2024.
- [15] Harper R Grimsley, George S Barron, Edwin Barnes, Sophia E Economou, and Nicholas J Mayhall. Adaptive, problem-tailored variational quantum eigensolver mitigates rough parameter landscapes and barren plateaus. *npj Quantum Information*, 9(1):19, 2023.
- [16] Shouzhen Gu, Rolando D Somma, and Burak Şahinoğlu. Fast-forwarding quantum evolution. *Quantum*, 5:577, 2021.
- [17] Artur F Izmaylov and Tzu-Ching Yen. How to define quantum mean-field solvable hamiltonians using lie algebras. *Quantum Science and Technology*, 6(4):044006, 2021.
- [18] Artur F Izmaylov, Tzu-Ching Yen, Robert A Lang, and Vladyslav Verteletskyi. Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method. *Journal of chemical theory and computation*, 16(1):190–195, 2019.
- [19] Navin Khaneja and Steffen J Glaser. Cartan decomposition of su (2n) and control of spin systems. *Chemical Physics*, 267(1-3):11–23, 2001.
- [20] Taehee Ko, Hyowon Park, and Sangkook Choi. Quantum random power method for ground state computation. *arXiv preprint arXiv:2408.08556*, 2024.
- [21] Efekan Kökü, Thomas Steckmann, Yan Wang, JK Freericks, Eugene F Dumitrescu, and Alexander F Kemper. Fixed depth hamiltonian simulation via cartan decomposition. *Physical Review Letters*, 129(7):070501, 2022.
- [22] Guoyin Li and Ting Kei Pong. Calculus of the exponent of kurdyka–lojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of computational mathematics*, 18(5):1199–1232, 2018.
- [23] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.
- [24] Guang Hao Low and Isaac L Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical review letters*, 118(1):010501, 2017.

- [25] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture. *arXiv preprint arXiv:1805.00675*, 2018.
- [26] Leonardo Novo, Juani Bermejo-Vega, and Raúl García-Patrón. Quantum advantage from energy measurements of many-body quantum systems. *Quantum*, 5:465, 2021.
- [27] Smik Patel, Tzu-Ching Yen, and Artur F Izmaylov. Extension of exactly-solvable hamiltonians using symmetries of lie algebras. *The Journal of Physical Chemistry A*, 128(20):4150–4159, 2024.
- [28] David Poulin, Matthew B Hastings, Dave Wecker, Nathan Wiebe, Andrew C Doherty, and Matthias Troyer. The Trotter step size required for accurate quantum simulation of quantum chemistry. *arXiv preprint arXiv:1406.4920*, 2014.
- [29] Ilya G Ryabinkin, Andrew J Jena, and Scott N Genin. Efficient construction of involutory linear combinations of anticommuting pauli generators for large-scale iterative qubit coupled cluster calculations. *Journal of chemical theory and computation*, 19(6):1722–1733, 2023.
- [30] Burak Şahinoğlu and Rolando D Somma. Hamiltonian simulation in the low-energy subspace. *npj Quantum Information*, 7(1):119, 2021.
- [31] Isaac D Smith, Maxime Caufrès, David T Stephen, and Hendrik Poulsen Nautrup. Optimally generating  $\mathfrak{su}(2^N)$  using pauli strings. *arXiv preprint arXiv:2408.03294*, 2024.
- [32] Rolando D Somma. Unitary circuit synthesis for tomography of generalized coherent states. *Journal of Mathematical Physics*, 60(11), 2019.
- [33] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, pages 217–228, 2019.
- [34] Ho Lun Tang, VO Shkolnikov, George S Barron, Harper R Grimsley, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum*, 2(2):020310, 2021.
- [35] Andrew Tranter, Peter J Love, Florian Mintert, Nathan Wiebe, and Peter V Coveney. Ordering of Trotterization: Impact on errors in quantum simulation of electronic structure. *Entropy*, 21(12):1218, 2019.
- [36] Linyun Wan, Jie Liu, Zhenyu Li, and Jinlong Yang. Variational-cartan quantum dynamics simulations of excitation dynamics, 2024.
- [37] Phillip Weinberg and Marin Bukov. Quspin: a python package for dynamics and exact diagonalisation of quantum many body systems part i: spin chains. *SciPost Physics*, 2(1):003, 2017.
- [38] David Wierichs, Maxwell West, Roy T. Forestano, M. Cerezo, and Nathan Killoran. Recursive cartan decompositions for unitary synthesis, 2025.
- [39] Roeland Wiersema, Efekan Kökcü, Alexander F Kemper, and Bojko N Bakalov. Classification of dynamical lie algebras of 2-local spin systems on linear, circular and fully connected topologies. *npj Quantum Information*, 10(1):110, 2024.