

Expanding Hardware-Efficiently Manipulable Hilbert Space via Hamiltonian Embedding

Jiaqi Leng^{*1,3,4}, Joseph Li^{*2,3}, Yuxiang Peng^{2,3}, and Xiaodi Wu^{2,3}

¹Department of Mathematics, University of Maryland, College Park, USA

²Department of Computer Science, University of Maryland, College Park, USA

³Joint Center for Quantum Information and Computer Science, University of Maryland

⁴Department of Mathematics and Simons Institute for the Theory of Computing, University of California, Berkeley

Many promising quantum applications depend on the efficient quantum simulation of an exponentially large sparse Hamiltonian, a task known as sparse Hamiltonian simulation, which is fundamentally important in quantum computation. Although several theoretically appealing quantum algorithms have been proposed for this task, they typically require a black-box query model of the sparse Hamiltonian, rendering them impractical for near-term implementation on quantum devices.

In this paper, we propose a technique named *Hamiltonian embedding*. This technique simulates a desired sparse Hamiltonian by embedding it into the evolution of a larger and more structured quantum system, allowing for more efficient simulation through hardware-efficient operations. We conduct a systematic study of this new technique and demonstrate significant savings in computational resources for implementing prominent quantum applications. As a result, we can now experimentally realize quantum walks on complicated graphs (e.g., binary trees, glued-tree graphs), quantum spatial search, and the simulation of real-space Schrödinger equations on current trapped-ion and neutral-atom platforms. Given the fundamental role of Hamiltonian evolution in the design of quantum algorithms, our technique markedly expands the horizon of implementable quantum advantages in the NISQ era.

Jiaqi Leng*: jiaqil@berkeley.edu

Xiaodi Wu: xiaodiwu@umd.edu

*Equal contribution.

Contents

1	Introduction	3
1.1	Review: quantum input models	6
1.2	Relevant work	7
2	Hamiltonian embedding	8
2.1	General formulation	8
2.2	Perturbative Hamiltonian embedding	9
2.3	Hamiltonian embedding of sparse matrices	10
2.4	Connection to quantum Hamiltonian complexity	12
3	Real-machine experiments	13
3.1	Hardware-efficient Hamiltonian models of quantum computers	13
3.2	Traversing the glued trees graph	14
3.3	Spatial search on square grids	16
3.4	Simulating real-space quantum dynamics	19
4	Open questions and discussion	21
A	Quantum simulation using Hamiltonian embedding	30
A.1	Proof of Theorem 1	30
A.2	Rules for building Hamiltonian embeddings	30
A.3	Perturbative Hamiltonian embedding	32
B	Hamiltonian embedding of sparse matrices	37
B.1	Band matrix	37
B.2	Banded circulant matrix	41
B.3	Arbitrary sparse matrices: one-hot embedding	43
C	Methodology of resource analysis	47
D	Quantum walk on graphs	48
D.1	Lattice graphs	48
D.2	Binary tree and glued trees graph	52
E	Spatial search on regular lattices	55
E.1	Embedding of the oracle Hamiltonian	56
E.2	Experiments on IonQ	57
F	Simulating quantum dynamics in the real space	59
F.1	Experiments on IonQ	60
F.2	Experiments on QuEra	64

1 Introduction

Quantum technology has achieved significant milestones, notably demonstrating quantum advantage over classical computers [1, 2], and the realization of error-corrected digital qubits [3]. Meanwhile, various quantum algorithms targeting practical applications have emerged, including solving large-scale linear systems [4], simulating linear and nonlinear dynamics [5, 6], and finding optimal solutions for mathematical optimization problems [7]. At the core of these quantum applications lies a fundamental subroutine named *sparse Hamiltonian simulation*, i.e., to compute the matrix function $f(A) = e^{-iAt}$ for a given sparse Hermitian matrix A . Over the past few decades, sparse Hamiltonian simulation has been a central topic in quantum computation research, and a rich variety of quantum algorithms (e.g., [8, 9, 10, 11, 12, 13]) have been developed for this task. While these algorithms achieve exponential quantum speedups over known classical ones, their implementation requires executing deep quantum circuits. The absence of fault tolerance in near-term devices hinders our ability to utilize these advanced quantum algorithms for solving classically intractable computational tasks in science and business [14].

Delving into the low-level implementation of quantum simulation algorithms reveals two prominent factors that contribute to the depth of quantum circuits. First, almost all sparse Hamiltonian simulation algorithms require a sophisticated quantum input model for the sparse matrix A , such as quantum oracles for sparse matrices [15, 8, 16], QRAM [17], or block-encodings [18, 13]. Implementing these input models usually introduces a significant gate count, even in relatively simple cases [19]. Second, in mapping a quantum algorithm to executable quantum gates, typical circuit compilation routines begin with representing the algorithm using standardized gates, such as CNOT and Toffoli. The gate sequence is further decomposed into native 1- and 2-qubit gates [20, 21, 22]. Failing to fully leverage the native programmability of the target quantum hardware, this compilation workflow often leads to significant practical overheads.

In this paper, we would like to explore an alternative approach to sparse Hamiltonian simulation that exploits both the sparsity structure of the input data and the resource efficiency within the underlying quantum hardware. To this end, we first introduce a Hamiltonian-based model of quantum algorithms that encompass all *hardware-efficient* operations in a specific quantum computer. Unlike quantum circuits, which act as a mathematical model agnostic to the underlying hardware, our new Hamiltonian model integrates low-level machine information to minimize the cost of end-to-end implementation. Based on this new model, we have developed a technique named *Hamiltonian embedding*, which enables us to simulate a sparse Hamiltonian as part of a larger quantum evolution that can be efficiently simulated on quantum hardware. With this technique, we can build input models directly with quantum Hamiltonians and perform sparse Hamiltonian simulations with extremely limited quantum resources.

Our Hamiltonian model of quantum computers is motivated by the low-level control logic of quantum hardware. In general, physically realizable quantum platforms, such as transmon qubits [23, 24], trapped ions [25, 26], neutral atoms [27], are described as quantum systems whose evolution is governed by a quantum Hamiltonian involving 1- and 2-body interactions. For example, the emulation of analog quantum simulators, such as quantum annealers [28, 29] and Rydberg atom arrays [30, 31], is governed by a system Hamiltonian composed of 1- and 2-local Pauli operators. In the case of digital quantum computers, their native continuously parameterized 1- and 2-qubit gates are typically specified by effective generating Hamiltonians.¹ We propose to use the following quantum Hamiltonian to model

¹For example, the arbitrary angle Mølmer-Sørensen gate $MS(\theta) = e^{-i(\theta/2)XX}$ in trapped-ion devices

hardware-efficient quantum operations:

$$H(t) = \sum_j \alpha_j(t) H_j + \sum_{j,k} \beta_{j,k}(t) H_{j,k}, \quad (1.1)$$

where H_j (or $H_{j,k}$) is a Hamiltonian on qubit j (or j and k) that corresponds to a native operation in specific hardware, $\alpha_j(t)$ and $\beta_{j,k}(t)$ are time-dependent functions. This Hamiltonian model can represent any implementable quantum circuits through piecewise-constant α_j and $\beta_{j,k}$.² On the other hand, any unitary operations generated by the Hamiltonian $H(t)$ with smooth time dependence can be readily implemented by a sequence of native gates via product formulas [34].

To illustrate the idea of Hamiltonian embedding, we consider a simple example. Suppose that we have two (time-independent) Hamiltonian operators A and H , and H contains A as a diagonal block such that $H = \text{diag}(A, *)$, where $*$ represents another (possibly different) Hamiltonian block that evolves independently within its own subspace and is therefore irrelevant to the simulation of A . Then, the time evolution generated by H is also block-diagonal,

$$e^{-iHt} = \begin{bmatrix} e^{-iAt} & 0 \\ 0 & * \end{bmatrix}, \quad (1.2)$$

where the upper left block is the time evolution of A . In other words, we can simulate a target Hamiltonian A by embedding it into a larger block-diagonal Hamiltonian H . Generalizing this intuition to “approximately” block-diagonal Hamiltonians, we have developed a formalism of Hamiltonian embedding with rigorous error analysis, see [Theorem 1](#). Now, if the parameters in the Hamiltonian model $H(t)$ are programmed in a way such that $H(t)$ embeds A , we end up with a *de facto* input model that allows efficient quantum simulation.³

In theory, a Hamiltonian embedding can be constructed for any sparse matrix, as detailed in [Section 2.3](#). For a general n -dimensional sparse matrix without specific structures, an embedding may require n qubits and $O(n^2)$ local interaction terms. In this case, Hamiltonian embedding could still offer polynomial speedups, although the actual advantage depends heavily on the problem class and requires case-by-case investigation. Fortunately, we managed to identify various scenarios, including high-dimensional graphs created through graph product operations and specific linear differential operators, where the Hamiltonian embedding can be constructed using quantum resources that scale logarithmically in the input size n . This leads to exponential quantum speedups using our methodology. When certain algebraic structures, such as addition, multiplication, composition, and tensor product, emerge in a sparse Hamiltonian A , we can decompose its Hamiltonian embedding into a small number of basic building blocks (see [Theorem 2](#)). We then provide six embedding schemes that can be employed to construct elementary building blocks of the full Hamiltonian embedding. These embedding schemes work for matrices with particular sparsity patterns (e.g., band, banded circulant, s -sparse), as detailed in [Section 2.3](#). It’s important to note that many of these embedding schemes are

are realized by engineering the effective Hamiltonian XX in the physical device for a variable time [32, 33].

²An implementable quantum circuit is composed of a sequence of native 1- and 2-qubit gates. Each execution of a native gate can be represented as a rectangular control signal in the time-dependent functions.

³An n -qubit Hamiltonian model has at most $O(n^2)$ 1- and 2-local component Hamiltonians, where each component Hamiltonian can be natively simulated in the associated quantum hardware. Then, we can use product formulas to simulate the time evolution generated by $H(t)$ with $\text{poly}(n)$ quantum resources.

already documented in existing literature, albeit under different names. Our contribution lies in developing a unifying formalism, which facilitates the systematic application of these schemes specifically for simulating large sparse Hamiltonian.

A key feature of Hamiltonian embedding is that it directly harnesses hardware-efficient operations to build the input model, which significantly reduces the quantum resources needed in Hamiltonian simulation tasks and quantum algorithms based on these. This technique enables us to implement several experiments on existing open-access cloud-based quantum computers, showcasing interesting quantum applications (see [Section 3](#)). In contrast, Hamiltonian simulation methods using traditional query models (such as quantum oracles or block-encodings) are impractical on these quantum computers. Even a single implementation of the query model would deplete the available quantum resources [19]. A detailed discussion on different quantum input models is provided in [Section 1.1](#). It is worth noting that our Hamiltonian embedding technique can also be adapted for analog quantum simulators,⁴ unlike existing sparse Hamiltonian simulation methods which are tailored exclusively for gate-based quantum computers. This adaptability broadens the possibilities for analog quantum computation.

In the experiments, we also provide comprehensive resource analyses (in terms of native gate count) to better understand the empirical scaling of Hamiltonian embeddings regarding various problem sizes. For comparison, we also estimate the quantum resources needed to perform the same Hamiltonian simulation tasks using the basic Pauli access model, which directly represents sparse Hamiltonians as a sum of Pauli operators, without embedding (see [Section 1.1](#) for details). While both Hamiltonian embedding and the Pauli access model use the Hamiltonian as inputs, the latter is unaware of the machine programmability and it utilizes the full Hilbert space to represent the sparse data.⁵ Due to these differences, we use the Pauli access model as a baseline to investigate the resource efficiency of our technique. It turns out that there always exists at least one Hamiltonian embedding that is more resource-efficient than the baseline in both asymptotic (i.e., scaling in system size) and non-asymptotic (i.e., actual gate count) metrics, as detailed in the panel B of [Figure 2](#), [3](#), and [4](#). Our findings indicate that Hamiltonian embedding, despite increasing the size of the global Hilbert space, provides a means to expand the hardware-efficiently manipulable Hilbert space. This leads to quantum simulations that are more resource-efficient compared to traditional hardware-agnostic approaches.

Contribution. Our main contributions are threefold. First, we propose to use the quantum Hamiltonian to model native operations in a quantum computer. This new model enables efficient Hamiltonian simulations without going through a hardware-agnostic compilation process. Second, we develop a new technique named Hamiltonian embedding for hardware-efficient sparse Hamiltonian simulation. We provide a general framework with rigorous error analysis and a flexible construction approach with concrete instances. Given some special structures in the sparse Hamiltonian simulation tasks, Hamiltonian embedding allows us to achieve up to exponential quantum speedups on a small fault-tolerant quantum computer. Last but not least, we showcase our technique to realize some interesting quantum applications that are almost infeasible with traditional Hamiltonian simulation methods. Comprehensive resource analysis shows that our methodology demonstrates both empirical and asymptotic advantages compared to the baseline method.

⁴An experimental demonstration with Rydberg atom arrays is provided in [Section 3.4](#).

⁵We remark that obtaining the Pauli access model for sparse matrices could require exponential classical pre-processing time. In the resource analysis, we assume the Pauli access model of a given A is already known.

Code Availability. The source codes of the real-machine experiments and resource analysis are available at <https://github.com/jiaqileng/hamiltonian-embedding>.

1.1 Review: quantum input models

In this subsection, we briefly review three commonly used quantum input models for Hamiltonian simulation and numerical linear algebra.

Sparse-input oracle. Let A be a matrix that is s -sparse, i.e., every row or column of A has at most s nonzero elements. The sparse-input oracles of A refer to a procedure (implemented by quantum circuits) that can perform the following mappings:

$$O_r: |i\rangle |k\rangle \rightarrow |i\rangle |r_{ik}\rangle, \quad O_c: |\ell\rangle |j\rangle \rightarrow |c_{\ell j}\rangle |j\rangle, \\ O_A: |i\rangle |j\rangle |0\rangle^{\otimes b} \rightarrow |i\rangle |j\rangle |a_{ij}\rangle,$$

where r_{ik} is the index for the k -th nonzero entry of the i -th row of A , $c_{\ell j}$ is the index for the ℓ -th nonzero entry of the j -th column of A , and a_{ij} is a b -bit binary description of the (i, j) -matrix element of A . This black-box query model was first considered by Aharonov and Ta-Shma [15], then has been widely assumed in quantum algorithms for Hamiltonian simulation [8, 9, 16, 12] and numerical linear algebra [35, 13]. However, even for sparse matrices with regular sparsity patterns and a small number of nonzero elements, implementing sparse-input oracles by efficient quantum circuits is a highly nontrivial task [19].

Block-encoding. A unitary U_A is a block-encoding of a matrix A if

$$\left\| A - \alpha(\langle 0|^{\otimes a} \otimes I)U_A(|0\rangle^{\otimes a} \otimes I) \right\| \leq \epsilon,$$

where α is a normalization factor, a is the number of ancilla qubits, and ϵ is an error parameter. Block-encoding arises naturally as an input model for algorithms based on quantum signal processing (QSP) [12] and the quantum singular value transformation (QSVT) [13]. However, it is not possible to construct circuits for block-encoding arbitrary sparse matrices with space and time complexity both logarithmic in the matrix dimension [36]. Efficient circuit constructions for block-encoding have only been studied for certain structured matrices [19, 37], but these constructions require sequences of multi-qubit controlled gates which are impractical for implementation on current devices. While the first steps towards implementing QSP have been demonstrated for a small-scale problem [38], due to the high cost of implementing a block-encoding, it is generally expected that the scalable implementation of QSP-based algorithms will only be possible in the deep fault-tolerant regime.

Pauli access model. The Pauli access model of an n -qubit Hamiltonian A assumes that A can be specified as a linear combination of Pauli operators Q_s ,

$$A = \sum_s a_s Q_s, \tag{1.3}$$

where $\{Q_s\}$ is the set of all n -qubit Pauli operators, and the coefficients a_s can be computed by $a_s = \frac{1}{2^n} \text{Tr}[AQ_s]$. In the literature, the representation (1.3) is sometimes referred to as the standard binary encoding of A , e.g., see [39, 40]. This input model has been considered in some variational quantum algorithms [41, 42] and randomized algorithms for linear algebra [43, 44]. Unlike sparse-input oracles and block-encodings, the Pauli access

model does not require a coherent quantum circuit implementation. However, for general sparse Hamiltonians, computing its Pauli operator decomposition is not scalable because the coefficient element a_s in (1.3) requires evaluation of the trace $\text{Tr}[AQ_s]$, which could run for an exponentially long time on classical computers. Also, evolving a Pauli operator that involves more than 2-site interactions (e.g., XYX) is expensive on gate-based quantum computers since the resulting unitary operators need to be decomposed to native 1- and 2-qubit gates. As a comparison, the construction of Hamiltonian embedding is scalable since it does not require computing the trace of large matrices. When the matrix A has certain sparsity patterns (e.g., band, banded circulant, etc.), the resulting Hamiltonian embedding only involves 2-site interactions and thus is readily implementable on physical hardware without further decomposition.

1.2 Relevant work

Hamming encoding in Quantum Hamiltonian Descent. Recently, Leng et al. [7] proposed a quantum optimization algorithm named Quantum Hamiltonian Descent (QHD). QHD addresses continuous optimization problems by simulating quantum evolution. In addition to providing a theoretical analysis of the quantum algorithm, the authors also developed an analog implementation of QHD, which they termed *Hamming encoding*. The Hamming encoding method can be seen as the initial instance of Hamiltonian embedding, in which the quantum algorithm is directly executed with a quantum Hamiltonian, rather than any existing quantum input models. Our current work expands this concept into a formalized approach, encompassing more constructions and a wider range of applications.

Encodings of quantum operators. Specific constructions of Hamiltonian embedding similar to our own have been extensively studied in different contexts. The facilitation or antiblockade phenomenon has been investigated via quantum dynamics confined to certain subspaces in arrays of Rydberg atoms [45, 46, 47]. The encoding of d -level quantum systems (i.e., qudits) within a multi-qubit Hamiltonian has also been explored for various encodings [39, 48, 49, 50], such as the standard binary, Gray, and one-hot codes. A software package named `mat2qubit` [40] automates the compilation of these encoding schemes, and in the case of the one-hot code, yields encoded operators identical to our (penalty-free) construction. Compared to our construction, these schemes generally lead to more complicated encoding operators because they completely disallow leakage to the orthogonal complement. The XY model has long been observed to give rise to quantum walk via the one-hot code [51], and restrictions to higher excitation subspaces have also been studied for the task of permuting a quantum state [52]. In the context of quantum adiabatic optimization [53, 54], the unary and one-hot codes have been studied to encode combinatorial optimization problems in the ground-energy subspace of certain penalty operators. While these optimization works do not consider the task of Hamiltonian simulation, a similar penalty operator is utilized in our constructions of Hamiltonian embeddings. From this perspective, Hamiltonian embedding can be viewed as a unifying framework that encompasses several well-studied encodings both with and without penalty.

Explicit construction of block-encodings for sparse matrices. The framework of quantum signal processing (QSP) has been shown to give an optimal algorithm for sparse Hamiltonian simulation [12]. In general, QSP-based algorithms require a block-encoding oracle of the Hamiltonian which is nontrivial to construct in general. Explicit circuit constructions of block-encodings have only been obtained for specific matrices [13, 19, 37].

In particular, [19] constructs circuits for block-encoding tri-diagonal and banded circulant matrices, for which we also consider Hamiltonian embeddings in this paper. Nevertheless, the circuits for block-encoding are constructed using multi-qubit controlled gates which require further decomposition into elementary one- and two-qubit gates. While the overall gate complexities are polylogarithmic in the matrix dimension (and thus considered efficient), the actual gate counts required even for a single oracle call are prohibitively expensive in practice. For instance, the circuit for block-encoding an 8×8 banded circulant matrix requires roughly 171 one-qubit gates and 114 two-qubit gates when compiling to Pauli- X , Pauli- Y , and XX rotations. On the other hand, Hamiltonian embedding provides an alternative approach that avoids such oracle constructions and enables near-term implementation on noisy quantum computers.

2 Hamiltonian embedding

2.1 General formulation

Let A be an n -dimensional Hermitian operator and H be a q -qubit operator.

Definition 1 (Hamiltonian embedding). Let $\eta, \epsilon > 0$ be positive scalars. We say H is a (q, η, ϵ) -**embedding** of A if there exists a subspace $\mathcal{S} \subset \mathbb{C}^{2^q}$ and a unitary operator U such that

1. $P_{\mathcal{S}}(U^\dagger H U) P_{\mathcal{S}^\perp} = 0$, i.e., $U^\dagger H U$ is block-diagonal in \mathcal{S} and \mathcal{S}^\perp ,
2. $\|I - U\| \leq \eta$, where I is the identity operator in \mathbb{C}^{2^q} ,
3. $\|(U^\dagger H U)|_{\mathcal{S}} - A\| \leq \epsilon$, where $(\cdot)|_{\mathcal{S}} := P_{\mathcal{S}}(\cdot)P_{\mathcal{S}}$.

We call the subspace \mathcal{S} as the **embedding subspace**.

By the definition, the operator H is approximately block-diagonal (up to a minor basis change U) with respect to the embedding subspace \mathcal{S} and its orthogonal complement \mathcal{S}^\perp . The target Hamiltonian A is embedded in the upper left block of $U^\dagger H U$ up to an additive error ϵ . Clearly, a q -qubit Hamiltonian H is a $(q, 0, 0)$ -embedding of itself. For sufficiently small η and ϵ , we show that the embedding Hamiltonian H simulates the time evolution generated by the target Hamiltonian A . The proof is given in [Appendix A.1](#).

Theorem 1 (Hamiltonian simulation with Hamiltonian embedding). *Suppose that H is a (q, η, ϵ) -embedding of A . Then, for a fixed evolution time $t \geq 0$, we have that*

$$\left\| \left(e^{-iHt} \right) \Big|_{\mathcal{S}} - e^{-iAt} \right\| \leq (2\eta\|H\| + \epsilon)t. \quad (2.1)$$

In addition, we find that a complicated Hamiltonian embedding can be built from simpler ones through the four composing rules, including addition, multiplication, composition, and tensor product. In what follows, we give an informal version of these rules. See [Appendix A.2](#) for a formal restatement and proof.

Theorem 2 (Rules for building Hamiltonian embeddings; informal).

1. (*Addition*) For $j = 1, 2$, let H_j be a (q, η, ϵ_j) -embedding of A_j , then $H_1 + H_2$ is a $(q, \eta, \epsilon_1 + \epsilon_2)$ -embedding of $A_1 + A_2$.
2. (*Multiplication*) Let H be a (q, η, ϵ_j) -embedding of A , then for a real scalar α , αH is a $(q, \eta, |\alpha|\epsilon)$ -embedding of αA .

3. (Composition) For $j = 1, 2$, let H_j be a $(q_j, \eta_j, \epsilon_j)$ -embedding of A_j , then $H_1 \otimes I + I \otimes H_2$ is a $(q_1 + q_2, \eta_1 + \eta_2, \epsilon_1 + \epsilon_2)$ -embedding of $A_1 \otimes I + I \otimes A_2$.
4. (Tensor product) For $j = 1, 2$, let H_j be a $(q_j, \eta_j, \epsilon_j)$ -embedding of A_j , then $H_1 \otimes H_2$ is a $(q_1 + q_2, \eta_1 + \eta_2, \|A_1\|\epsilon_2 + \|A_2\|\epsilon_1 + \epsilon_1\epsilon_2)$ -embedding of $A_1 \otimes A_2$.

2.2 Perturbative Hamiltonian embedding

As we have seen, a target Hamiltonian A can be simulated with a block-diagonal embedding Hamiltonian. However, identifying such a corresponding block-diagonal embedding could be non-trivial for many sparse Hamiltonians A arising from real-world applications. Here we give an explicit construction of Hamiltonian embedding based on perturbation theory.

First, we assume there is a q -qubit quantum operator Q and a subspace \mathcal{S} such that $Q|_{\mathcal{S}} = A$. We express the operator Q in a block-matrix form by projecting it down to the subspaces \mathcal{S} and \mathcal{S}^\perp , respectively,

$$Q = \begin{bmatrix} A & R^\dagger \\ R & B \end{bmatrix}, \quad (2.2)$$

where $A = P_{\mathcal{S}}QP_{\mathcal{S}}$, $R = P_{\mathcal{S}^\perp}QP_{\mathcal{S}}$, $B = P_{\mathcal{S}^\perp}QP_{\mathcal{S}^\perp}$. Since the off-diagonal block R is not necessarily zero, we can not directly simulate A by evolving the Hamiltonian Q because in general $P_{\mathcal{S}}e^{-iQt}P_{\mathcal{S}} \neq e^{-iAt}$. This means that a quantum state initialized in the subspace \mathcal{S} could be driven out of this subspace in the course of quantum evolution.

To suppress the leakage from the embedding subspace \mathcal{S} , we introduce another q -qubit operator H^{pen} as the *penalty Hamiltonian*. We assume H^{pen} has an n -fold degenerate ground-energy subspace \mathcal{S} with the ground energy being zero, i.e., the least eigenvalue of H^{pen} is zero. For a fixed positive number $g > 0$, we define the following quantum operator,

$$H = gH^{\text{pen}} + Q. \quad (2.3)$$

Similarly, this new operator H can be expressed in a block-matrix form (we denote $C = P_{\mathcal{S}^\perp}H^{\text{pen}}P_{\mathcal{S}^\perp}$ and $G = B + gC$):

$$H = \begin{bmatrix} A & R^\dagger \\ R & B + gC \end{bmatrix} = \underbrace{\begin{bmatrix} A & 0 \\ 0 & G \end{bmatrix}}_{\text{diagonal}} + \underbrace{\begin{bmatrix} 0 & R^\dagger \\ R & 0 \end{bmatrix}}_{\text{off-diagonal}}, \quad (2.4)$$

The Hamiltonian H decomposes into an off-diagonal part and a diagonal part. For sufficiently large $g > 0$, there is a gap between the spectrum of A and that of G and the width of this gap is proportional to g , i.e.,

$$\Delta = \lambda_{\min}(G) - \lambda_{\max}(A) > 0, \quad \Delta \sim g. \quad (2.5)$$

When g is large enough such that $\|R\|/g \ll 1$, the off-diagonal part in (2.4) can be treated as a perturbative term. In this case, we prove that H is an embedding of A , as shown in the following theorem. A formal statement and the proof are provided in [Appendix A.3](#).

Theorem 3 (Perturbative Hamiltonian embedding; informal version of [Theorem 8](#)). *Let the operators H , A , and R be the same as above. For sufficiently large $g > 0$, the Hamiltonian H is a (q, η, ϵ) -embedding of A , where $\eta \sim \|R\|/g$, $\epsilon \sim \|R\|^2/g$.*

Given that $\|R\| \leq \|H\|$, this result immediately implies that the simulation error using perturbative Hamiltonian embedding is of the order $\mathcal{O}(\|R\|\|H\|t/g)$, where the error bound follows from [Theorem 1](#). This error bound can be improved to $\mathcal{O}(\|R\|^2t/g)$ by leveraging the particular operator splitting structure as shown in [\(2.4\)](#), see [Theorem 10](#). In practice, the embedding subspace \mathcal{S} is often much smaller than the full Hilbert space; consequently, it is often sufficient to assume $\|R\| = O(1)$. Therefore, to achieve a simulation error $\delta > 0$, we need to choose a penalty coefficient $g = \mathcal{O}(t/\delta)$. However, this choice of g would incur a $\text{poly}(1/\delta)$ overhead in the overall gate complexity when we implement the Hamiltonian simulation on gate-based quantum computers using standard product formulas. Fortunately, the penalty Hamiltonian H^{pen} is usually fast-forwardable, i.e., the time-evolution operator $e^{-iH^{\text{pen}}t}$ can be simulated using quantum resources that scale sub-linear in t . In this case, we find the unfavorable $\text{poly}(1/\delta)$ overhead can be mitigated by utilizing interaction-picture quantum simulation algorithms like continuous qDRIFT [\[55\]](#). See [Appendix A.3.3](#) for a detailed discussion.

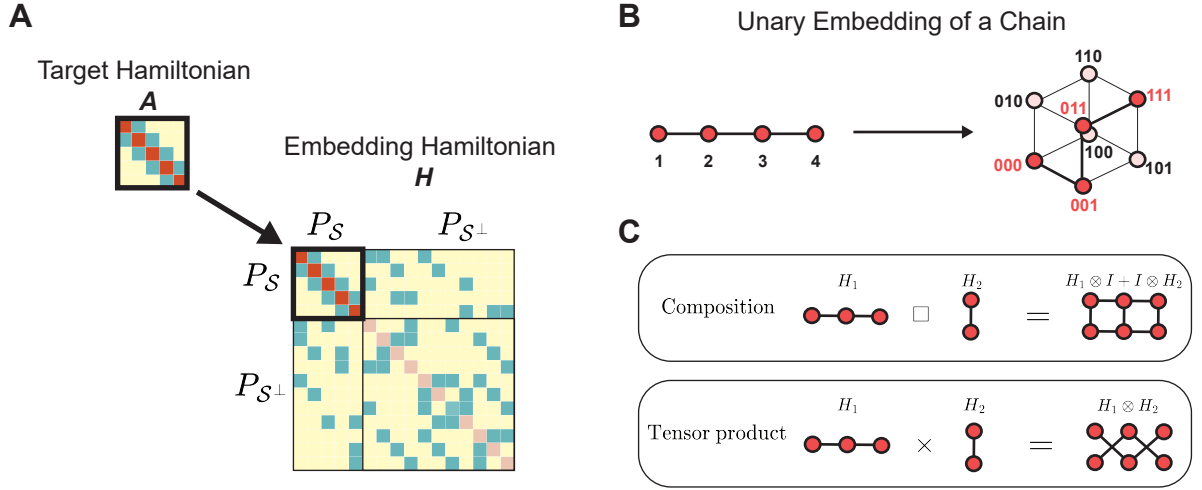


Figure 1: **Hamiltonian embedding.** **A.** Illustration of perturbative Hamiltonian embedding. The dynamics of a target Hamiltonian A are embedded within a subspace \mathcal{S} in a larger hardware Hilbert space. With respect to \mathcal{S} and \mathcal{S}^\perp , H is block-diagonal with A being the upper-left block located by the projector P_S . **B.** Graphical illustration of Hamiltonian embedding for the case of a chain (i.e. tridiagonal matrix) using the unary embedding. From a graph theoretic perspective, a path graph is located as a subgraph of a hypercube graph. Red vertices indicate basis states in the embedding subspace \mathcal{S} , while pink vertices indicate basis states in \mathcal{S}^\perp . **C.** Visualization of composition and tensor product rules for constructing Hamiltonian embeddings. When considering the graph of nonzero entries of a Hamiltonian, these rules are analogous to two different notions of graph products (see [Remark 1](#) in the appendix for more details).

2.3 Hamiltonian embedding of sparse matrices

For Hermitian matrices with certain special sparsity structures, we can construct explicit Hamiltonian embeddings for them. In what follows, we provide 6 different Hamiltonian embedding schemes for 3 types of sparse Hermitian matrices:

1. **Band matrix:** an n -by- n matrix A is a band matrix of bandwidth d if $A_{i,j} = 0$ for any $i, j = 1, \dots, n$ such that $|i - j| > d$.
2. **Banded circulant matrix:** an n -by- n matrix A is a banded circulant matrix of bandwidth d if (i) in the first row, $A_{1,j} = 0$ for any $j = d + 2, \dots, n - d$, and (ii) all

row vectors are composed of the same elements and each row vector is rotated one element to the right relative to the preceding row vector.

3. ***s*-sparse matrices**: an n -by- n matrix A is s -sparse if each row/column of A has at most s non-zero elements.

In [Table 1](#), we list all the embedding schemes discussed in this paper that can be applied to simulate at least one type of sparse Hermitian matrices. Among our embedding schemes, the unary and one-hot encodings are well-known [[53](#), [54](#), [39](#), [40](#), [51](#)], while the others have not been systematically studied in the existing literature. Note that, except for the last item (“Penalty-free one-hot”), all embeddings are perturbative Hamiltonian embeddings and thus require a penalty Hamiltonian.⁶ The “Max. weight” column shows the maximal weight of the Pauli operators involved in the corresponding embedding Hamiltonian (d represents the bandwidth of the target Hamiltonian A). Full details of these embedding schemes can be found in [Appendix B](#).

Embedding scheme	Applications	Max. weight	Details
Unary	band	$\max(d, 2)$	Appendix B.1.1
Antiferromagnetic	band	$\max(d, 2)$	Appendix B.1.2
Circulant unary	banded circulant	$\max(d, 2)$	Appendix B.2.1
Circulant antiferromagnetic	banded circulant	$\max(d, 2)$	Appendix B.2.2
One-hot	band, banded circulant, s -sparse	2	Appendix B.3.1
Penalty-free one-hot	band, banded circulant, s -sparse	2	Appendix B.3.2

Table 1: Six Hamiltonian embeddings for sparse Hamiltonian simulation

A notable feature of these embedding schemes is that their maximal weight depends on the special sparsity pattern of the target Hamiltonian A . In particular, for band matrices (or banded circulant matrices) with bandwidth $d \leq 2$, the corresponding unary/antiferromagnetic (or circulant unary/circulant antiferromagnetic) embeddings have a maximal weight of 2, which means these embedding Hamiltonians can be directly simulated using native gates. This is also true for one-hot/penalty-free one-hot embeddings that encode arbitrary s -sparse matrices.

We also note that all these schemes utilize only an exponentially small subspace of the full Hilbert space for embedding. Therefore, direct use of these schemes to build Hamiltonian embeddings does not lead to meaningful quantum speedups in Hamiltonian simulation. To achieve exponential quantum speedups, one strategy is to first decompose the target sparse Hamiltonian using the *composition* and *tensor product* rules in [Theorem 2](#), and then use the embedding schemes listed in [Table 1](#) to construct elementary building blocks of size $O(1)$. Using this strategy, we can build Hamiltonian embeddings for symmetric/Hermitian matrices arising from graph theory and differential equations with logarithmic quantum resources, as detailed in [Section 3](#).

Now, we provide some concrete examples of the embedding of a small sparse matrix.

⁶For perturbative Hamiltonian embedding, the simulation error can be made arbitrarily small by increasing the penalty coefficient g . Thus, for simplicity, we do not explicitly specify the tuple (g, η, ϵ) for each Hamiltonian embedding listed in the table.

We consider a 5-by-5 tridiagonal matrix A given by

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \quad (2.6)$$

which is the Laplacian matrix of a 5-node chain graph. In [Table 2](#), we list the operators Q and H^{pen} for embedding A using various embedding schemes (with more details available in [Appendix D](#)). In this paper, we use little-endian ordering, meaning that we enumerate the bit of a string *from right to left*. Here, $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ are the Pauli- X and Pauli- Z matrices, and $\hat{n} = \frac{1}{2}(I - Z)$. The overall embedding Hamiltonian is $H = gH^{\text{pen}} + Q$, where $g > 0$ is a sufficiently large penalty coefficient. The embedding subspace \mathcal{S} depends on the embedding scheme. For sufficiently large $g > 0$, [Theorem 3](#) implies that $e^{-iAt} \approx P_{\mathcal{S}} e^{-iH_A t} P_{\mathcal{S}}$.

Embedding scheme	Q	H^{pen}
Unary	$-\hat{n}_1 + \hat{n}_4 + \sum_{j=1}^4 X_j$	$-\sum_{j=1}^3 Z_{j+1} Z_j + Z_1 - Z_4$
Antiferromagnetic	$-\hat{n}_1 - \hat{n}_4 + \sum_{j=1}^4 X_j$	$\sum_{j=1}^3 Z_{j+1} Z_j + Z_1 + Z_4$
One-hot	$\left(-\hat{n}_1 - \hat{n}_5 - 2 \sum_{j=2}^4 \hat{n}_j\right) + \sum_{j=1}^4 X_{j+1} X_j$	$\left(\sum_{j=1}^5 \hat{n}_j - 1\right)^2$
Penalty-free one-hot	$\left(-\hat{n}_1 - \hat{n}_5 - 2 \sum_{j=2}^4 \hat{n}_j\right) + \frac{1}{2} \sum_{j=1}^4 (X_{j+1} X_j + Y_{j+1} Y_j)$	0

Table 2: Hamiltonian embeddings of the tridiagonal matrix A in [\(2.6\)](#).

While there could be several possible Hamiltonian embedding schemes for a fixed target Hamiltonian A , we remark that there is no single criterion to determine which one would perform the best. We provide a few aspects to compare various Hamiltonian embeddings in practice. First, the Hamiltonian embedding must match the hardware programmability. For example, the antiferromagnetic embedding scheme fits well with Rydberg atom arrays, see [Appendix F.2.2](#) for details. Second, we want to use the Hamiltonian embedding with minimal simulation error. The simulation error could come from the perturbative construction of an embedding and/or a specific Hamiltonian simulation algorithm (e.g., a Trotter formula, continuous qDRIFT, etc.).

2.4 Connection to quantum Hamiltonian complexity

Hamiltonian embedding is closely related to previously studied notions of simulating one Hamiltonian by another Hamiltonian, including the isometry-based definition of *simulation* in [\[56\]](#) used to study the complexity of 2-local stoquastic Hamiltonians, and *Hamiltonian encodings* used to show the universality of certain spin-lattice models [\[57, 58\]](#).

The simulation introduced in [\[56\]](#) quantifies how close are two quantum Hamiltonians in terms of their low-energy spectrum. It is defined as an isometry transformation and its application has been limited to stoquastic local Hamiltonians [\[59\]](#), while our Hamiltonian embedding applies to a broader class of sparse matrices.

A Hamiltonian encoding (aka, encoding transformation) is a map that encodes a Hamiltonian H into some other Hamiltonian H' . Specifically, this map needs to fulfill a few basic requirements, including the preservation of locality, spectrum, and real-linearity [57]. Hamiltonian encoding is more general than simulation since an encoding does not need to be an isometry. The construction of Hamiltonian encodings heavily utilizes perturbative gadgets [60, 61, 62], a theoretical tool originally developed to prove hardness results in Hamiltonian complexity theory.

Compared to [57], our definition of Hamiltonian embedding (Definition 1) more closely resembles the *simulation* defined in [56], in the sense that both do not impose any locality-related restriction on the encoding/embedding maps. Technically speaking, our perturbative Hamiltonian embedding can be viewed as an explicit construction of perturbative gadgets, but two major differences distinguish our work from prior arts. First, existing techniques are almost exclusively applied to (local) quantum Hamiltonians arising from many-body physics (i.e., bosons, fermions, and local spin/qudit Hamiltonians), while our focus is on the simulation of sparse matrices. Second, Hamiltonian encodings often involve complicated basis change, while our Hamiltonian embedding preserves the energy spectrum and dynamical evolution without introducing a nontrivial basis change. This allows us to measure the embedded system using a subset of computational basis. However, this favorable feature is achieved at a cost of universality. The construction technique in [57] can be applied to arbitrary local Hamiltonians, while we only provide explicit constructions of Hamiltonian embedding for certain families of sparse Hermitian matrices.

3 Real-machine experiments

We conduct experiments to demonstrate the use of Hamiltonian embeddings for computational tasks, including (continuous-time) quantum walk on graphs, spatial search, and simulating real-space quantum dynamics. We construct Hamiltonian embeddings for each task and deploy them on current digital and analog quantum computers, including IonQ’s ion trap systems [63] and QuEra’s neutral atom systems [64]. We also exhibit the efficiency and scalability of our approach over the conventional Pauli access approach (i.e., the standard binary encoding) through a detailed resource analysis. If given the Pauli decomposition of a Hamiltonian, the Pauli access model enables a straightforward approach to Hamiltonian simulation via product formulas. On the other hand, the sparse-input and block-encoding input models require coherent circuit implementations of oracles which are highly nontrivial even for structured matrices. While there has been recent progress for constructing block-encodings [19, 37], the overhead associated with these schemes make them applicable only in the fault-tolerant regime. Consequently, we compare our embedding schemes with the standard binary encoding, which serves as a more reasonable baseline. More details on resource analysis are available in Appendix C.

3.1 Hardware-efficient Hamiltonian models of quantum computers

In this subsection, we give the hardware-efficient Hamiltonian models for the IonQ and QuEra quantum computers. For both devices, their Hamiltonian models can be formulated as

$$H(t) = \sum_j \alpha_j(t) H_j + \sum_{j,k} \beta_{j,k}(t) H_{j,k},$$

but with different native 1- and 2-qubit component Hamiltonians.

The IonQ Aria-1 quantum computer natively supports the GPI gate, GPI2 gate [65], virtual-Z gate, and arbitrary angle MS (Mølmer-Sørensen) gate [66]. The hardware-efficient Hamiltonian model for the IonQ Aria-1 device is composed of the following 1- and 2-qubit components:

$$H_j: aX + bY + cZ, \quad H_{j,k}: (\cos(\phi_1)X_j + \sin(\phi_1)Y_j) \otimes (\cos(\phi_2)X_k + \sin(\phi_2)Y_k), \quad (3.1)$$

where a, b, c, ϕ_1, ϕ_2 can be any real-valued scalars.

The QuEra Aquila quantum computer is an analog quantum simulator. It allows users to program certain parameters, including the Rabi frequency, local detuning, and atom-atom distance, in the effective Hamiltonian describing the Rydberg atom arrays. More details on the QuEra device are provided in [Appendix F.2.2](#). We can formulate the abstract model for the QuEra Aquila device using the following 1- and 2-qubit component Hamiltonians:

$$H_j: aX + bY + cZ, \quad H_{j,k}: \alpha Z \otimes Z, \quad (3.2)$$

where a, b, c are real-valued scalars. The parameter α is engineered by Rydberg interactions and thus must be non-negative.

3.2 Traversing the glued trees graph

We first demonstrate the simulation of continuous-time quantum walk (CTQW) on undirected graphs [67]. In this section, we focus on the simulation of quantum walk on a glued trees graph, but the embedding and simulation of CTQW for other graphs are presented in [Appendix D](#). The problem of traversing the glued trees graph is proven to be efficiently solvable for quantum computers while computationally hard for classical computers in the oracle setting [68]. Although our experiment simulates the desired quantum dynamics, this result itself does not imply achieving exponential quantum speedup on NISQ devices because the Hamiltonian embedding is not built using logarithmic quantum resources.

In the existing literature, CTQW on glued trees has been experimentally realized only in very limited settings. [69] used a photonic chip to simulate a one-dimensional quantum walk reduced from the CTQW on glued trees. An earlier photonic implementation [70] realizes quantum walk on a hexagonal variant of the glued trees graph. Implementations of CTQW have been explored more extensively for other graphs on various experimental platforms, such as NMR [71], neutral atoms [72, 73], superconducting qubits [74], and photonic processors [75]. However, these implementations are hardware-specific and generally require qubits proportional to the order (i.e. number of vertices) of the graph, thus unable to accommodate other families of graphs. The work in [76, 77] simulates CTQW in a photon interference experiment, but the desired evolution operators are calculated classically to configure linear optical circuits. Discrete-time quantum walks have also been realized on a trapped-ion quantum processor [78], which uses a dense encoding (thus being space-efficient) but is essentially hard-coded for a specific choice of the initial state.

In [Figure 2A](#), we illustrate a 14-node glued trees graph $G = (V, E)$ containing two balanced binary trees. Here, V is the set of vertices, and E is the set of all edges. The CTQW on this graph is described by the time-evolution operator e^{-itA} , where t is the evolution time, and A is a 14-by-14 real symmetric matrix representing the adjacency matrix of the graph:

$$A_{jk} = \begin{cases} 1, & j \neq k, (j, k) \in E, \\ 0, & j \neq k, (j, k) \notin E. \end{cases} \quad (3.3)$$

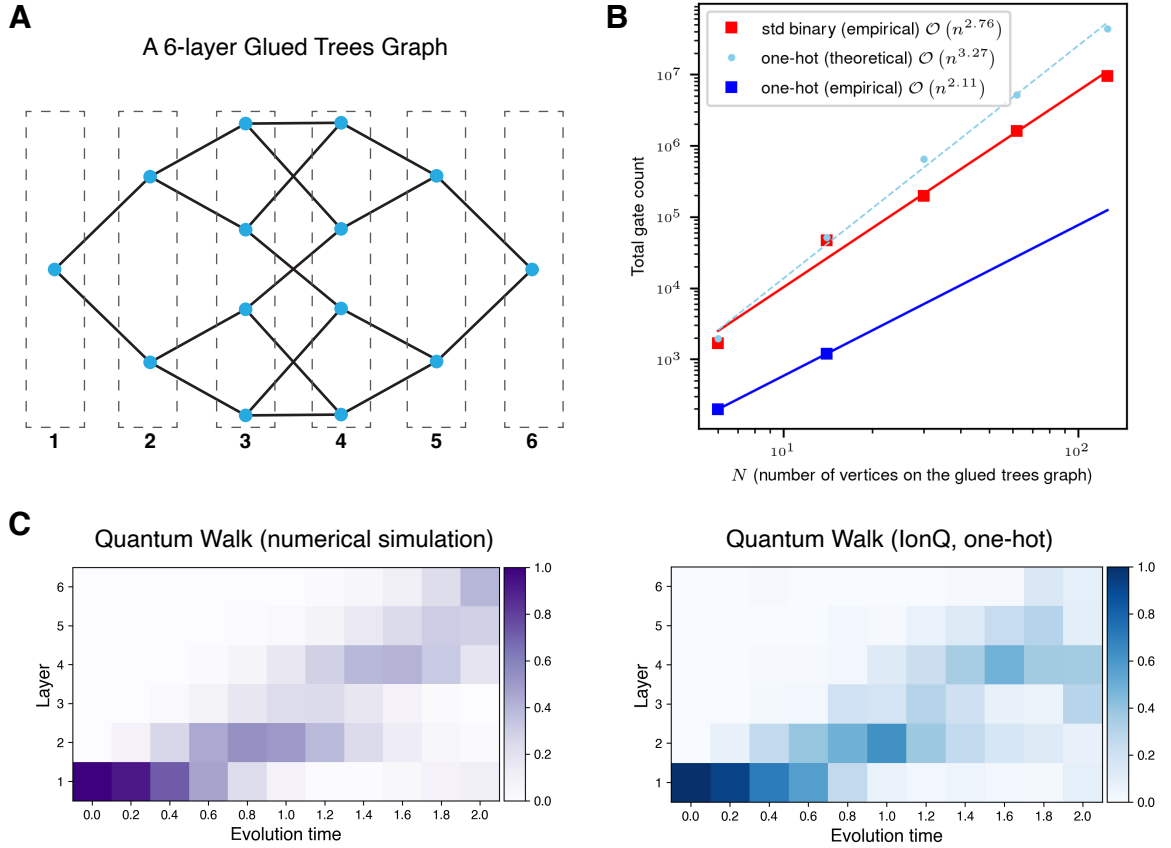


Figure 2: **Quantum walk on glued trees graphs.** **A.** A 6-layer glued trees graph with 14 vertices. **B.** Gate counts of simulating CTQW on N -vertex glued trees graphs using standard binary encoding and one-hot embedding. Both the theoretical (worst-case) upper bound and the empirical data suggest that the one-hot embedding has better asymptotic scaling. **C.** Propagation of the quantum wave packet on the 6-layer glued trees graph. Wave packet clustered on each layer. Left: numerical simulation; Right: real-machine results.

To simulate the Hamiltonian A , we construct the following 14-qubit penalty-free one-hot embedding (details in [Appendix D.2](#)),

$$H = \sum_{(j,k) \in E} \frac{1}{2} (X_j X_k + Y_j Y_k), \quad (3.4)$$

where the embedding subspace \mathcal{S} is spanned by the one-hot codewords $\{|0..01\rangle, |0..10\rangle, \dots, |10..0\rangle\}$. In other words, \mathcal{S} is the single-excitation subspace in the 14-qubit Hilbert space. It is readily verified that \mathcal{S} is an invariant subspace of H and $H|_{\mathcal{S}} = A$, so the simulation of A is embedded within the dynamics of H .

[Figure 2B](#) shows the total gate counts for the circuit implementation via standard binary encoding and Hamiltonian embedding (i.e., penalty-free one-hot). The total gate counts (represented by solid dots) are estimated such that the simulation error is suppressed to a fixed accuracy. The results from extrapolation indicate that in terms of both exact and asymptotic measures, Hamiltonian embedding outperforms standard binary encoding. A detailed discussion on the resource analysis for this experiment is available in [Appendix D.2.4](#).

Encoding/embedding	# of qubits	# of 1-qubit gates	# of 2-qubit gates
Standard binary	4	6088	932
Penalty-free one-hot	14	1	160

Table 3: 1- and 2-qubit gate counts for simulating CTQW on a 14-node glued trees graph. The accuracy is chosen to correspond to the Trotter number $r = 4$ for the one-hot code as used in the real-machine experiment.

We implement the Hamiltonian embedding as shown in (3.4) on the IonQ Aria-1 processor to simulate the CTQW on the 14-node glued trees graph, where we employ the randomized first-order Trotter formula to decompose the simulation into circuits with IonQ’s native gate set. We initialize the quantum state with a single walker (starting from the entrance node in layer 1) at time $t = 0$ and simulate the propagation of the quantum wave function through the layers over time. The real-machine and numerical simulation results are illustrated in Figure 2C. In both subplots, a clear pattern of population migration over time is observed. The real-machine result depicts a strong similarity to the numerical simulation, while a slightly shifted population distribution is witnessed near the end time $T = 2$, possibly caused by the accumulated device noise. In Table 3, we list the empirical resource usage (qubit and gate counts) for traversing the glued trees graph using the one-hot code as well as an estimate of the resources required by the standard binary code. While the standard binary code uses fewer qubits, the gate counts needed are an order of magnitude larger. The savings in gate count for the penalty-free one-hot embedding enables the simulation of sparse Hamiltonians which would otherwise be infeasible on current devices.

3.3 Spatial search on square grids

Another prominent application for quantum computers is to find a target entry in a database, a task known as *quantum search*. The famous Grover’s algorithm [79] tackles the quantum search task with $O(\sqrt{M})$ queries to entries in a database of size M , while classical algorithms require at least $\Omega(M)$ queries, showcasing a quadratic quantum speedup. In this section, we implement a quantum search algorithm proposed by Childs and Goldstone [80] for a structured database (i.e., *spatial search*). The search spaces are two-dimensional square grids with N^2 entries. By the composition rule of Hamiltonian embedding, we can implement the aforementioned spatial search algorithm using only $2N$ qubits. In general, spatial search over a d -dimensional grid (with N^d entries) can be implemented using dN qubits by composing Hamiltonian embeddings (see Appendix A.2).

There are only a few experimental demonstrations of spatial search documented in the literature. Some photonic-based implementations have been shown in [81, 82], while these proposals only work for specific graphs and thereby lack programmability. A proof-of-principle demonstration of two-dimensional spatial search was performed using neutral atoms [73], where N^2 atoms were used to represent all entries in the search space. In this regard, our implementation of spatial search serves as the first demonstration of spatial search on a lattice that effectively exploits the matrix structure of the Hamiltonian. The use of Hamiltonian embedding significantly reduces the required qubit and gate counts, thereby enabling a digital implementation of 2-dimensional spatial search that has not been demonstrated in any previous literature.

In 2004, Childs and Goldstone [80] proposed a quantum search algorithm via quantum walk. Given a search space represented by a graph $G = (V, E)$ and a marked entry $v \in V$,

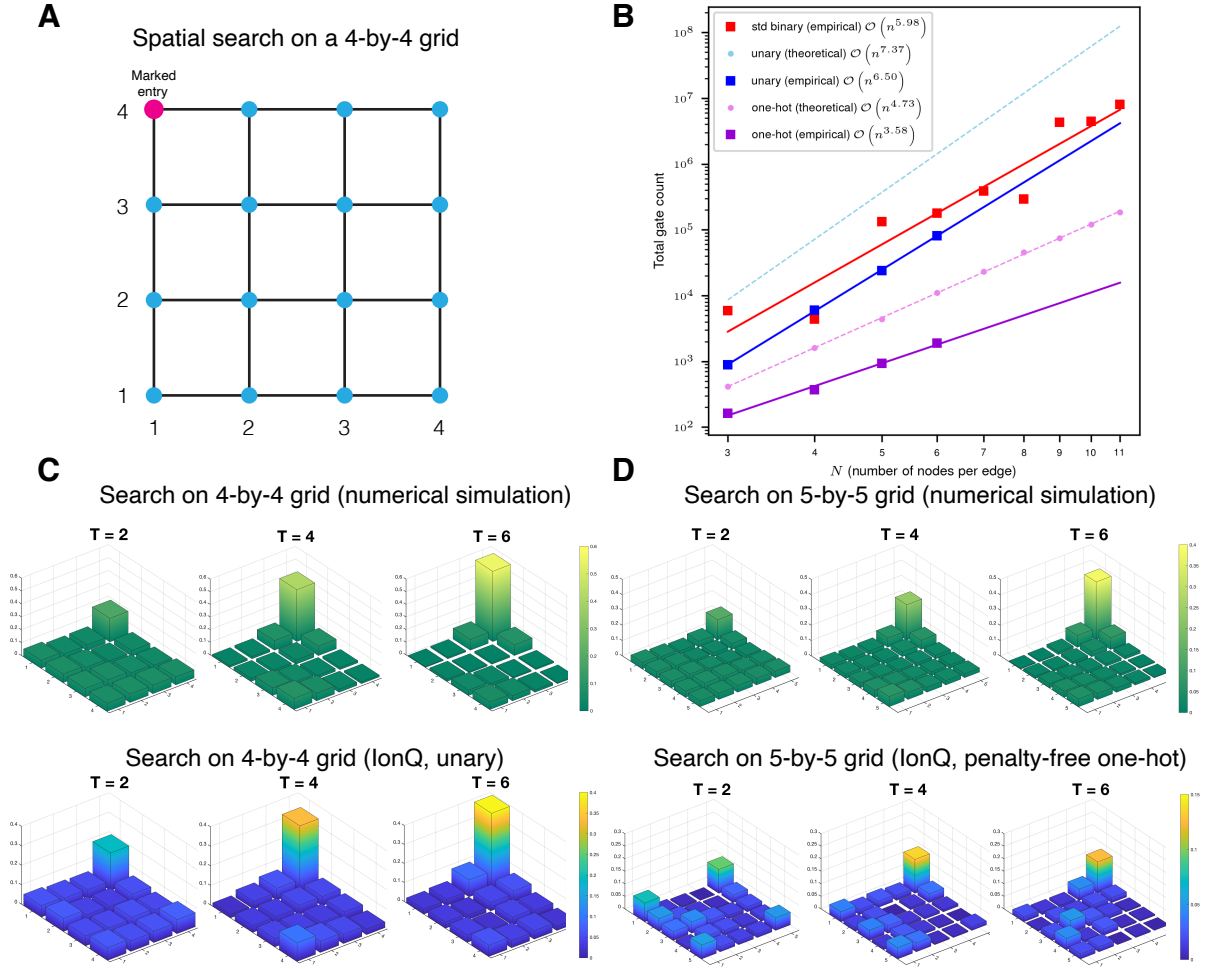


Figure 3: **Spatial search by quantum walk.** **A.** Illustration of a spatial search problem on a 4-by-4 grid. **B.** Gate counts of implementing spatial search on N -by- N grids. The penalty-free one-hot embedding has the best asymptotic scaling in both the worst-case estimate and empirical extrapolation. **C.** Quantum distribution at $T = 2, 4, 6$ shows the algorithm solves the 4-by-4 spatial search problem. Top: numerical simulation; Bottom: results obtained on lonQ Aria-1 with unary embedding. **D.** Quantum distribution at $T = 2, 4, 6$ shows the algorithm solves the 5-by-5 spatial search problem. Top: numerical simulation; Bottom: results obtained on lonQ Aria-1 with penalty-free one-hot embedding.

their quantum algorithm requires simulating the following Hamiltonian,

$$H = -\gamma L - H_v, \quad (3.5)$$

where L is the Laplacian of the graph G , $H_v = |v\rangle\langle v|$ is a projector onto the marked entry v (known as the *oracle* Hamiltonian), and $\gamma > 0$ is a parameter minimizing the spectral gap in H . When the search space is a two-dimensional grid graph (with N^2 entries (j, k) for $j, k = 1, \dots, N$) and the marked entry is $\mathbf{v} = (v_x, v_y)$, the graph Laplacian reads

$$L = L_1 \otimes I_N + I_N \otimes L_1, \quad (3.6)$$

where L_1 is the graph Laplacian of a N -node one-dimensional chain graph (an example for $N = 5$ is given in (2.6)), and I_N is the N -by- N identity operator. Meanwhile, the oracle Hamiltonian is given by

$$H_{\mathbf{v}} = |v_x, v_y\rangle\langle v_x, v_y| = |v_x\rangle\langle v_x| \otimes |v_y\rangle\langle v_y|. \quad (3.7)$$

In [Figure 3A](#), we illustrate a spatial search problem on a 4-by-4 grid with the marked entry at the upper left corner. By indexing the entries from the lower left corner, the oracle Hamiltonian for the marked entry is $H_v = |1, 4\rangle\langle 1, 4|$.

To implement the quantum algorithm by Childs and Goldstone to solve the two-dimensional spatial search problem, we need to simulate the target Hamiltonian H given in [\(3.5\)](#), which is of dimension N^2 . Since H is a sparse Hamiltonian, we could in principle construct a one-hot embedding using N^2 qubits. However, the decomposition structure in the graph Laplacian [\(3.6\)](#) and the tensor product structure in the oracle Hamiltonian [\(3.7\)](#) allow us to build a more compact Hamiltonian embedding using Hamiltonian embeddings of L_1 , $|v_x\rangle\langle v_x|$, and $|v_y\rangle\langle v_y|$. Notably, for both the unary and one-hot Hamiltonian embeddings, we only require $O(N)$ qubits and $O(N)$ 2-local Pauli operators. More details of the construction for spatial search (including the case of periodic boundaries) may be found in [Appendix E](#).

[Figure 3B](#) shows the total gate counts in the circuit implementation via standard binary encoding and Hamiltonian embeddings (i.e., unary, penalty-free one-hot). The extrapolation results suggest that the penalty-free one-hot embedding (both worst-case and empirical estimates) outperforms the standard binary encoding in both exact and asymptotic performance measures. However, the unary embedding appears to use more elementary gates than the standard binary encoding, potentially due to the large penalty coefficient. More discussions on the resource analysis for this experiment can be found in [Appendix E.2.2](#).

Experimental results for two-dimensional spatial search on 4-by-4 and 5-by-5 grids are shown in panels C and D in [Figure 3](#). We choose the unary embedding for the 4-by-4 experiment and the penalty-free one-hot embedding for the 5-by-5 experiment. More details of the experiment setup, including the state preparation procedure, are provided in [Appendix E.2](#). The results show that our implementations via Hamiltonian embedding on the IonQ device are in good agreement with the expected quantum dynamics described in the algorithm. In [Table 4](#) and [Table 5](#), we list the empirical resource usage for

Encoding/embedding	# of qubits	# of 1-qubit gates	# of 2-qubit gates
Standard binary	4	831	123
Unary	6	132	114

Table 4: Resource counts for simulating spatial search on a 4×4 lattice.

Encoding/embedding	# of qubits	# of 1-qubit gates	# of 2-qubit gates
Standard binary	6	26100	4464
Penalty-free one-hot	10	22	181

Table 5: Resource counts for simulating spatial search on a 5×5 lattice.

spatial search on 4×4 and 5×5 lattices, respectively. Compared to our embedding schemes, the standard binary encoding requires orders of magnitude more gates for the same target accuracy. Consequently, Hamiltonian embedding demonstrates a significant resource savings that enables the implementation of quantum search on current quantum computers.

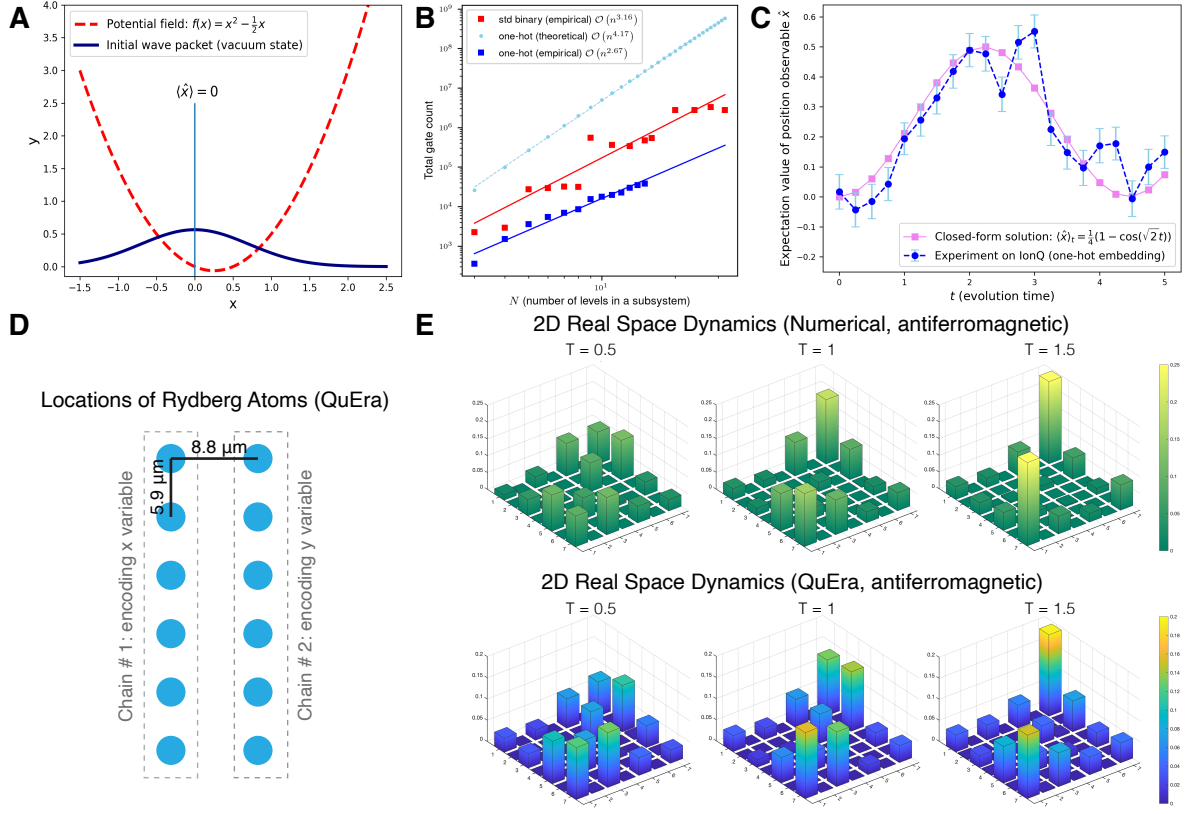


Figure 4: **Simulating real-space quantum dynamics.** **A.** The potential field and initial state in the lonQ experiment. **B.** Resource count of simulating real-space quantum dynamics using an N -level truncated Hamiltonian. The empirical data suggests that one-hot embedding has a better asymptotic scaling than the standard binary encoding. **C.** Expectation value of the position observable \hat{x} depicted as a function of evolution time t . Experiment data obtained from lonQ Aria-1. **D.** Locations of the 12 Rydberg atoms in the experiment of simulating quantum dynamics in a 2D space. **E.** Measuring the 2D real-space quantum dynamics at $T = 1, 1.5, 2$. Top: numerical simulation; Bottom: experiment results obtained on QuEra Aquila.

3.4 Simulating real-space quantum dynamics

The real-space quantum dynamics are governed by the time-dependent Schrödinger equation over the d -dimensional Euclidean space,

$$i \frac{\partial}{\partial t} \Psi(t, x) = \left[-\frac{1}{2} \nabla^2 + f(x) \right] \Psi(t, x), \quad (3.8)$$

$$\text{subject to } \Psi(0, x) = \Psi_0(x), \quad (3.9)$$

where $\Psi(t, x) : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{C}$ is the quantum wave function, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is a potential field, and the initial state is given by $\Psi_0(x)$.

Although physically relevant systems are of dimension 3, high-dimensional Schrödinger equations find ubiquitous applications for simulating multi-particle systems in condensed matter physics [83] and quantum chemistry [84, 85]. Also, some quantum algorithms for numerical optimization require the simulation of Schrödinger equations in high-dimensional Euclidean spaces [86, 87, 7]. Several quantum algorithms for real-space quantum simulation have been proposed, e.g., [88, 89, 90, 91, 92, 93, 94], while most of them require fully fault-tolerant quantum computers that are currently out of reach. Recently, Chang et al. [95]

utilize variants of Gray code to reformulate the continuous-space Schrödinger equation as spin Hamiltonians, leading to exponentially better space complexity over classical methods. However, the spin operators in [95] involve 3-body interaction terms and thus require further decomposition in the implementation. In our experiment, the different choices of embedding schemes allow us to represent the Schrödinger equation using 1- and 2-body spin operators, eliminating the Pauli compilation overhead.

We showcase the flexibility and versatility of the Hamiltonian embedding technique by implementing the real-space quantum simulation task on two different experimental platforms: a trapped-ion quantum processor (IonQ Aria-1) and a programmable Rydberg atom array (QuEra Aquila).

On the trapped-ion processor, we simulate a 1-dimensional Schrödinger equation with a quadratic potential field and a Gaussian initial state,

$$f(x) = x^2 - \frac{1}{2}x, \quad \Psi_0(x) = \left(\frac{1}{2\pi}\right)^{1/4} e^{-\frac{x^2}{4}},$$

as illustrated in Figure 4A. We use a truncated Fock space method with $N = 5$ levels to discretize the Hamiltonian operator and obtain a finite-dimensional sparse Hamiltonian. Next, we employ the penalty-free one-hot embedding to build the corresponding embedding Hamiltonian

$$H_{\text{real-space}} = \frac{1}{2}Q_{\hat{p}^2}^{\text{one-hot}} - \frac{1}{2}Q_{\hat{x}^2}^{\text{one-hot}} + \frac{1}{2}Q_{\hat{x}}^{\text{one-hot}}, \quad (3.10)$$

where $Q_{\hat{p}^2}^{\text{one-hot}}$, $Q_{\hat{x}^2}^{\text{one-hot}}$, and $Q_{\hat{x}}^{\text{one-hot}}$ are all 2-local Hamiltonians. More details of the truncation method and embedding are discussed in Appendix F.1.2.

In Figure 4B, we estimate and compare the gate counts required in the circuit implementation for the standard binary encoding and the penalty-free one-hot embedding. The extrapolation based on empirical data shows that our Hamiltonian embedding implementation uses fewer elementary gates and has slightly better asymptotic scaling in terms of system size, while the theoretical (i.e., worst-case) asymptotic scaling of our embedding does not show an advantage.

We simulate the embedding Hamiltonian $H_{\text{real-space}}$ on the IonQ Aria-1 processor and compute the expectation value of the position observable \hat{x} as a function of time as shown Figure 4C. Notably, computing \hat{x} only requires measurements in the x -basis when using Hamiltonian embedding. The same measurements using the standard binary code would require performing measurements in several different bases corresponding to the Pauli decomposition of \hat{x} , which demonstrates another advantage of Hamiltonian embedding. The experimental data matches the closed-form solution and we observe a full oscillation from $t = 0$ to $t = 5$. More details regarding the experiment setup and results are provided in Appendix F.1.3. Furthermore, we list in Table 6 the empirical resource usage for simulating the real-space Schrödinger equation with the one-hot code as well as an estimate of the resources required by the standard binary code. Although we consider only a 1-dimensional case in this example, the one-hot code demonstrates a significant advantage in gate count. In particular, the one-hot code only requires a single 1-qubit gate (for state preparation), while the standard binary code requires over 1800 single-qubit gates, which would not be feasible on current quantum hardware.

We also use programmable Rydberg atom arrays to simulate a 2-dimensional Schrödinger equation. First, we apply the finite difference method to discretize the Hamiltonian operator, yielding a finite-dimensional Hamiltonian,

$$\hat{H} = -\frac{1}{2}(D \otimes I + I \otimes D) + U, \quad (3.11)$$

Encoding/embedding	# of qubits	# of 1-qubit gates	# of 2-qubit gates
Standard binary	3	1826	220
Penalty-free one-hot	5	1	154

Table 6: 1- and 2-qubit gate counts for simulating the real-space Schrödinger equation with $N = 5$ levels. The accuracy is chosen to correspond to the Trotter number $r = 11$ for the one-hot code as used in the real-machine experiment.

where D is an N -by- N tridiagonal matrix representing the finite-difference discretization of the second-order differential operator $\frac{\partial^2}{\partial x^2}$, I is the N -by- N identity matrix and U is a N^2 -by- N^2 diagonal matrix corresponding to the potential field $f(x, y)$. The native Hamiltonian of Rydberg atom arrays allows us to form a Hamiltonian embedding of (3.11) using the antiferromagnetic scheme, while other schemes (such as unary) are not possible because the Rydberg interaction coefficients must be positive (see Appendix F.2.2 for details). The desired Hamiltonian embedding can be realized by arranging the atoms into two chains as shown in Figure 4D, where each chain represents an individual continuous variable (x for chain 1, y for chain 2). Note that the lack of local detuning in QuEra Aquila poses a significant restriction on the shape of the potential field. In our experiment, the *effective* potential field U and the penalty Hamiltonian are both engineered by the pairwise Rydberg interactions between atoms. As shown in Figure 4E, the QuEra-implemented quantum simulation results have good agreement with the numerical simulation. The quantum distributions are obtained using 1000 shots/measurements per time step (for $T = 0.5, 1, 2$). The details of the finite difference discretization, together with the construction and implementation of Hamiltonian embedding, are presented in Appendix F.2.

4 Open questions and discussion

In this work, we propose the Hamiltonian embedding technique for hardware-efficient sparse Hamiltonian simulation. This approach has the potential to achieve exponential quantum speedups when a large sparse Hamiltonian can be efficiently decomposed using the rules as described in Theorem 2. We then identify several instances from graph theory, combinatorial optimization, and differential equations where Hamiltonian embeddings can be constructed in poly-logarithmic time. It remains an interesting question whether Hamiltonian embeddings can be efficiently constructed for practical problems with less regular structures, potentially leading to practical quantum advantages even with limited quantum resources.

Our analysis of the perturbative Hamiltonian embedding (see Theorem 3) indicates a non-negligible simulation error for a large evolution time, which can only be suppressed by a large penalty coefficient. This is because we only utilize the first-order Schrieffer–Wolff theory to craft perturbative Hamiltonian embedding. It is of interest to explore if higher-order Schrieffer–Wolff theories could lead to more efficient Hamiltonian embeddings.

In the resource analysis, the empirical scalings of Hamiltonian embedding are usually much better than the results suggested by the theoretical (worst-case) analysis. This may suggest that we need new analytical tools to better understand the resource efficiency of Hamiltonian embedding.

Our real-machine demonstrations of Hamiltonian embedding are limited to small-scale toy model problems, and the on-device experiment results do not fully match the ideal numerical simulation. This could be caused by the accumulated machine noise. With the evolution of quantum hardware in the next few years, we are excited about the opportunity

to further explore the practical quantum advantages that Hamiltonian embedding can offer in broad application domains, such as simulating non-quantum linear dynamics or even nonlinear dynamics at a large scale.

Last but not least, while the primary focus of this work is to expand the implementable quantum advantages for near-term (NISQ) devices, the Hamiltonian embedding technique also opens new possibilities for quantum input models and could be highly relevant even for fault-tolerant quantum computation. For example, the construction of Hamiltonian embeddings relies on the Pauli input model, which can be naturally leveraged by quantum algorithms such as quantum signal processing or quantum singular value transformation without the need to explicitly construct block encodings [96]. Moreover, Hamiltonian embedding appears to offer a trade-off between system connectivity (local interactions) and space complexity, an active area of research in quantum compilation and circuit routing.

Acknowledgment

We thank Alexey Gorshkov, Lei Fan, Lexing Ying, and Lin Lin for helpful discussions and insightful feedback. This work was partially funded by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing under Award Number DE-SC0020273, the Air Force Office of Scientific Research under Grant No. FA95502110051, the U.S. National Science Foundation grant CCF-1816695 and CCF-1942837 (CAREER), and a Sloan research fellowship. JL is partially supported by DOE QSA Grant No. FP00010905, a Simons Quantum Postdoctoral Fellowship, and a Simons Investigator award through Grant No. 825053.

References

- [1] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. “Quantum supremacy using a programmable superconducting processor”. *Nature* **574**, 505–510 (2019).
- [2] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. “Quantum computational advantage using photons”. *Science* **370**, 1460–1463 (2020).
- [3] Dolev Bluvstein, Simon J Evered, Alexandra A Geim, Sophie H Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al. “Logical quantum processor based on reconfigurable atom arrays”. *Nature* **626**, 58–65 (2024).
- [4] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. “Quantum algorithm for linear systems of equations”. *Phys. Rev. Lett.* **103**, 150502 (2009).
- [5] Dong An, Andrew M. Childs, and Lin Lin. “Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters” (2023). [arXiv:2312.03916](https://arxiv.org/abs/2312.03916).
- [6] Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. “Efficient quantum algorithm for dissipative non-linear differential equations”. *Proceedings of the National Academy of Sciences* **118**, e2026805118 (2021).
- [7] Jiaqi Leng, Ethan Hickman, Joseph Li, and Xiaodi Wu. “Quantum Hamiltonian Descent” (2023). [arXiv:2303.01471](https://arxiv.org/abs/2303.01471).
- [8] Dominic W Berry, Graeme Ahokas, Richard Cleve, and Barry C Sanders. “Efficient quantum algorithms for simulating sparse Hamiltonians”. *Communications in Mathematical Physics* **270**, 359–371 (2007).
- [9] Dominic W. Berry and Andrew M. Childs. “Black-box Hamiltonian simulation and unitary implementation”. *Quantum Information & Computation* **12**, 29–62 (2012).
- [10] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. “Exponential improvement in precision for simulating sparse Hamiltonians”. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing (STOC '14)*. Page 283–292. New York, NY, USA (2014). Association for Computing Machinery.
- [11] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. “Simulating Hamiltonian dynamics with a truncated Taylor series”. *Phys. Rev. Lett.* **114**, 090502 (2015).
- [12] Guang Hao Low and Isaac L. Chuang. “Optimal Hamiltonian simulation by quantum signal processing”. *Phys. Rev. Lett.* **118**, 010501 (2017).
- [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 (STOC '19)*. Page 193–204. Phoenix, AZ, USA (2019). Association for Computing Machinery.
- [14] Michael E. Beverland, Prakash Murali, Matthias Troyer, Krysta M. Svore, Torsten Hoefler, Vadym Kliuchnikov, Guang Hao Low, Mathias Soeken, Aarthi Sundaram, and Alexander Vaschillo. “Assessing requirements to scale to practical quantum advantage” (2022). [arXiv:2211.07629](https://arxiv.org/abs/2211.07629).

- [15] Dorit Aharonov and Amnon Ta-Shma. “Adiabatic quantum state generation and statistical zero knowledge”. In Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing (STOC ’03). [Page 20–29](#). San Diego, CA, USA (2003). Association for Computing Machinery.
- [16] Andrew M. Childs and Robin Kothari. “Simulating sparse Hamiltonians with star decompositions”. In Wim van Dam, Vivien M. Kendon, and Simone Severini, editors, Theory of Quantum Computation, Communication, and Cryptography. [Pages 94–103](#). Berlin, Heidelberg (2011). Springer Berlin Heidelberg.
- [17] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. “Quantum random access memory”. [Phys. Rev. Lett. **100**, 160501](#) (2008).
- [18] Guang Hao Low and Nathan Wiebe. “Hamiltonian simulation in the interaction picture” (2019). [arXiv:1805.00675](#).
- [19] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. “Explicit quantum circuits for block encodings of certain sparse matrices”. [SIAM Journal on Matrix Analysis and Applications **45**, 801–827](#) (2024).
- [20] Qiskit contributors. “Qiskit: An open-source framework for quantum computing” (2023).
- [21] Anders Sørensen and Klaus Mølmer. “Quantum computation with ions in thermal motion”. [Phys. Rev. Lett. **82**, 1971–1974](#) (1999).
- [22] SA Caldwell, N Didier, CA Ryan, EA Sete, A Hudson, P Karalekas, R Manenti, MP da Silva, R Sinclair, E Acala, et al. “Parametrically activated entangling gates using transmon qubits”. [Phys. Rev. Appl. **10**, 034050](#) (2018).
- [23] Andreas Wallraff, David I Schuster, Alexandre Blais, Luigi Frunzio, R-S Huang, Johannes Majer, Sameer Kumar, Steven M Girvin, and Robert J Schoelkopf. “Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics”. [Nature **431**, 162–167](#) (2004).
- [24] Michel H Devoret and Robert J Schoelkopf. “Superconducting circuits for quantum information: an outlook”. [Science **339**, 1169–1174](#) (2013).
- [25] Rainer Blatt and Christian F Roos. “Quantum simulations with trapped ions”. [Nature Physics **8**, 277–284](#) (2012).
- [26] C. Monroe, W. C. Campbell, L.-M. Duan, Z.-X. Gong, A. V. Gorshkov, P. W. Hess, R. Islam, K. Kim, N. M. Linke, G. Pagano, P. Richerme, C. Senko, and N. Y. Yao. “Programmable quantum simulations of spin systems with trapped ions”. [Rev. Mod. Phys. **93**, 025001](#) (2021).
- [27] Loïc Henriët, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. “Quantum computing with neutral atoms”. [Quantum **4**, 327](#) (2020).
- [28] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, Richard Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. “Quantum annealing with manufactured spins”. [Nature **473**, 194–198](#) (2011).
- [29] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose. “Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor”. [Phys. Rev. B **82**, 024511](#) (2010).
- [30] Sepehr Ebadi, Tout T Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho,

- et al. “Quantum phases of matter on a 256-atom programmable quantum simulator”. [Nature](#) **595**, 227–232 (2021).
- [31] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin. “Quantum optimization of maximum independent set using Rydberg atom arrays”. [Science](#) **376**, 1209–1215 (2022).
- [32] Klaus Mølmer and Anders Sørensen. “Multiparticle entanglement of hot trapped ions”. [Phys. Rev. Lett.](#) **82**, 1835–1838 (1999).
- [33] E. Solano, R. L. de Matos Filho, and N. Zagury. “Deterministic Bell states and measurement of the motional state of two trapped ions”. [Phys. Rev. A](#) **59**, R2539–R2543 (1999).
- [34] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. “Theory of Trotter error with commutator scaling”. [Phys. Rev. X](#) **11**, 011020 (2021).
- [35] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. “Quantum algorithm for systems of linear equations with exponentially improved dependence on precision”. [SIAM Journal on Computing](#) **46**, 1920–1950 (2017).
- [36] Xiao-Ming Zhang and Xiao Yuan. “Circuit complexity of quantum access models for encoding classical data”. [npj Quantum Information](#) **10**, 42 (2024).
- [37] Christoph Sünderhauf, Earl Campbell, and Joan Camps. “Block-encoding structured matrices for data input in quantum computing”. [Quantum](#) **8**, 1226 (2024).
- [38] Yuta Kikuchi, Conor Mc Keever, Luuk Coopmans, Michael Lubasch, and Marcello Benedetti. “Realization of quantum signal processing on a noisy quantum computer”. [npj Quantum Information](#) **9**, 93 (2023).
- [39] Nicolas PD Sawaya, Tim Menke, Thi Ha Kyaw, Sonika Johri, Alán Aspuru-Guzik, and Gian Giacomo Guerreschi. “Resource-efficient digital quantum simulation of d -level systems for photonic, vibrational, and spin- s Hamiltonians”. [npj Quantum Information](#) **6**, 49 (2020).
- [40] Nicolas PD Sawaya. “mat2qubit: A lightweight pythonic package for qubit encodings of vibrational, bosonic, graph coloring, routing, scheduling, and general matrix problems” (2022). [arXiv:2205.09776](#).
- [41] Carlos Bravo-Prieto, Ryan LaRose, Marco Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J Coles. “Variational quantum linear solver”. [Quantum](#) **7**, 1188 (2023).
- [42] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. “Variational algorithms for linear algebra”. [Science Bulletin](#) **66**, 2181–2188 (2021).
- [43] Kianna Wan, Mario Berta, and Earl T. Campbell. “Randomized quantum algorithm for statistical phase estimation”. [Phys. Rev. Lett.](#) **129**, 030503 (2022).
- [44] Samson Wang, Sam McArdle, and Mario Berta. “Qubit-efficient randomized quantum algorithms for linear algebra”. [PRX Quantum](#) **5**, 020324 (2024).
- [45] Matteo Marcuzzi, Jiří Minář, Daniel Barredo, Sylvain De Léséleuc, Henning Labuhn, Thierry Lahaye, Antoine Browaeys, Emanuele Levi, and Igor Lesanovsky. “Facilitation dynamics and localization phenomena in Rydberg lattice gases with position disorder”. [Phys. Rev. Lett.](#) **118**, 063606 (2017).
- [46] Maike Ostmann, Matteo Marcuzzi, Jiří Minář, and Igor Lesanovsky. “Synthetic lattices, flat bands and localization in Rydberg quantum simulators”. [Quantum Science and Technology](#) **4**, 02LT01 (2019).
- [47] Fangli Liu, Zhi-Cheng Yang, Przemyslaw Bienias, Thomas Iadecola, and Alexey V. Gorshkov. “Localization and criticality in antiblockaded two-dimensional Rydberg atom arrays”. [Phys. Rev. Lett.](#) **128**, 013603 (2022).

- [48] Nicolas P. D. Sawaya, Gian Giacomo Guerreschi, and Adam Holmes. “On connectivity-dependent resource requirements for digital quantum simulation of d -level particles”. In 2020 IEEE International Conference on Quantum Computing and Engineering (QCE). Pages 180–190. (2020).
- [49] Thi Ha Kyaw, Tim Menke, Sukin Sim, Abhinav Anand, Nicolas P.D. Sawaya, William D. Oliver, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. “Quantum computer-aided design: Digital quantum simulation of quantum processors”. *Phys. Rev. Appl.* **16**, 044042 (2021).
- [50] Thi Ha Kyaw, Micheline B Soley, Brandon Allen, Paul Bergold, Chong Sun, Victor S Batista, and Alán Aspuru-Guzik. “Boosting quantum amplitude exponentially in variational quantum algorithms”. *Quantum Science and Technology* **9**, 01LT01 (2023).
- [51] Matthias Christandl, Nilanjana Datta, Tony C. Dorlas, Artur Ekert, Alastair Kay, and Andrew J. Landahl. “Perfect transfer of arbitrary states in quantum spin networks”. *Phys. Rev. A* **71**, 032312 (2005).
- [52] Claudio Albanese, Matthias Christandl, Nilanjana Datta, and Artur Ekert. “Mirror inversion of quantum states in linear registers”. *Phys. Rev. Lett.* **93**, 230502 (2004).
- [53] Nicholas Chancellor. “Domain wall encoding of discrete variables for quantum annealing and QAOA”. *Quantum Science and Technology* **4**, 045004 (2019).
- [54] Stuart Hadfield, Zihui Wang, Bryan O’gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. “From the quantum approximate optimization algorithm to a quantum alternating operator ansatz”. *Algorithms* **12**, 34 (2019).
- [55] Dominic W Berry, Andrew M Childs, Yuan Su, Xin Wang, and Nathan Wiebe. “Time-dependent Hamiltonian simulation with L^1 -norm scaling”. *Quantum* **4**, 254 (2020).
- [56] Sergey Bravyi and Matthew Hastings. “On complexity of the quantum Ising model”. *Communications in Mathematical Physics* **349**, 1–45 (2017).
- [57] Toby S. Cubitt, Ashley Montanaro, and Stephen Piddock. “Universal quantum Hamiltonians”. *Proceedings of the National Academy of Sciences* **115**, 9497–9502 (2018).
- [58] Leo Zhou and Dorit Aharonov. “Strongly universal Hamiltonian simulators” (2021). [arXiv:2102.02991](https://arxiv.org/abs/2102.02991).
- [59] Sergey Bravyi, David P. DiVincenzo, Roberto I. Oliveira, and Barbara M. Terhal. “The complexity of stoquastic local Hamiltonian problems” (2007). [arXiv:quant-ph/0606140](https://arxiv.org/abs/quant-ph/0606140).
- [60] Julia Kempe, Alexei Kitaev, and Oded Regev. “The complexity of the local Hamiltonian problem”. *SIAM journal on computing* **35**, 1070–1097 (2006).
- [61] Roberto Oliveira and Barbara M. Terhal. “The complexity of quantum spin systems on a two-dimensional square lattice” (2008). [arXiv:quant-ph/0504050](https://arxiv.org/abs/quant-ph/0504050).
- [62] Stephen P. Jordan and Edward Farhi. “Perturbative gadgets at arbitrary orders”. *Phys. Rev. A* **77**, 062329 (2008).
- [63] IonQ. “IonQ API documentation” (2023). <https://docs.ionq.com/>.
- [64] Jonathan Wurtz, Alexei Bylinskii, Boris Braverman, Jesse Amato-Grill, Sergio H. Cantu, Florian Huber, Alexander Lukin, Fangli Liu, Phillip Weinberg, John Long, Sheng-Tao Wang, Nathan Gemelke, and Alexander Keesling. “Aquila: Quera’s 256-qubit neutral-atom quantum computer (version 1.0)” (2023). [arXiv:2306.11727](https://arxiv.org/abs/2306.11727).
- [65] Kenneth Wright, Kristin M Beck, Sea Debnath, JM Amini, Y Nam, N Grzesiak, J-S Chen, NC Panti, M Chmielewski, C Collins, et al. “Benchmarking an 11-qubit quantum computer”. *Nature communications* **10**, 5464 (2019).

- [66] IonQ. “Getting started with native gates” (2023). <https://ionq.com/docs/getting-started-with-native-gates>.
- [67] Andrew M Childs. “On the relationship between continuous-and discrete-time quantum walk”. *Communications in Mathematical Physics* **294**, 581–603 (2010).
- [68] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. “Exponential algorithmic speedup by a quantum walk”. In Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing (STOC ’03). Page 59–68. San Diego, CA, USA (2003). Association for Computing Machinery.
- [69] Zi-Yu Shi, Hao Tang, Zhen Feng, Yao Wang, Zhan-Ming Li, Jun Gao, Yi-Jun Chang, Tian-Yu Wang, Jian-Peng Dou, Zhe-Yong Zhang, Zhi-Qiang Jiao, Wen-Hao Zhou, and Xian-Min Jin. “Quantum fast hitting on glued trees mapped on a photonic chip”. *Optica* **7**, 613–618 (2020).
- [70] Hao Tang, Carlo Di Franco, Zi-Yu Shi, Tian-Shen He, Zhen Feng, Jun Gao, Ke Sun, Zhan-Ming Li, Zhi-Qiang Jiao, Tian-Yu Wang, et al. “Experimental quantum fast hitting on hexagonal graphs”. *Nature Photonics* **12**, 754–758 (2018).
- [71] Jiangfeng Du, Hui Li, Xiaodong Xu, Mingjun Shi, Jihui Wu, Xianyi Zhou, and Rongdian Han. “Experimental implementation of the quantum random-walk algorithm”. *Phys. Rev. A* **67**, 042316 (2003).
- [72] Philipp M Preiss, Ruichao Ma, M Eric Tai, Alexander Lukin, Matthew Rispoli, Philip Zupancic, Yoav Lahini, Rajibul Islam, and Markus Greiner. “Strongly correlated quantum walks in optical lattices”. *Science* **347**, 1229–1233 (2015).
- [73] Aaron W. Young, William J. Eckner, Nathan Schine, Andrew M. Childs, and Adam M. Kaufman. “Tweezer-programmable 2D quantum walks in a Hubbard-regime lattice”. *Science* **377**, 885–889 (2022).
- [74] Ming Gong, Shiyu Wang, Chen Zha, Ming-Cheng Chen, He-Liang Huang, Yulin Wu, Qingling Zhu, Youwei Zhao, Shaowei Li, Shaojun Guo, et al. “Quantum walks on a programmable two-dimensional 62-qubit superconducting processor”. *Science* **372**, 948–952 (2021).
- [75] Dengke Qu, Lei Xiao, Kunkun Wang, Xiang Zhan, and Peng Xue. “Experimental investigation of equivalent Laplacian and adjacency quantum walks on irregular graphs”. *Phys. Rev. A* **105**, 062448 (2022).
- [76] Xiaogang Qiang, Yizhi Wang, Shichuan Xue, Renyou Ge, Lifeng Chen, Yingwen Liu, Anqi Huang, Xiang Fu, Ping Xu, Teng Yi, et al. “Implementing graph-theoretic quantum algorithms on a silicon photonic quantum walk processor”. *Science Advances* **7**, eabb8375 (2021).
- [77] Yizhi Wang, Yingwen Liu, Junwei Zhan, Shichuan Xue, Yuzhen Zheng, Ru Zeng, Zhihao Wu, Zihao Wang, Qilin Zheng, Dongyang Wang, et al. “Large-scale full-programmable quantum walk and its applications” (2022). [arXiv:2208.13186](https://arxiv.org/abs/2208.13186).
- [78] C Huerta Alderete, Shivani Singh, Nhung H Nguyen, Daiwei Zhu, Radhakrishnan Balu, Christopher Monroe, CM Chandrashekar, and Norbert M Linke. “Quantum walks and Dirac cellular automata on a programmable trapped-ion quantum computer”. *Nature communications* **11**, 3720 (2020).
- [79] Lov K. Grover. “Quantum mechanics helps in searching for a needle in a haystack”. *Phys. Rev. Lett.* **79**, 325–328 (1997).
- [80] Andrew M. Childs and Jeffrey Goldstone. “Spatial search by quantum walk”. *Phys. Rev. A* **70**, 022314 (2004).
- [81] Claudia Benedetti, Dario Tamascelli, Matteo G.A. Paris, and Andrea Crespi. “Quan-

- tum spatial search in two-dimensional waveguide arrays”. *Phys. Rev. Applied* **16**, 054036 (2021).
- [82] Dengke Qu, Samuel Marsh, Kunkun Wang, Lei Xiao, Jingbo Wang, and Peng Xue. “Deterministic search on star graphs via quantum walks”. *Phys. Rev. Lett.* **128**, 050501 (2022).
- [83] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan. “Low-depth quantum simulation of materials”. *Phys. Rev. X* **8**, 011044 (2018).
- [84] Ivan Kassal, Stephen P. Jordan, Peter J. Love, Masoud Mohseni, and Alán Aspuru-Guzik. “Polynomial-time quantum algorithm for the simulation of chemical dynamics”. *Proceedings of the National Academy of Sciences* **105**, 18681–18686 (2008).
- [85] Ryan Babbush, Dominic W Berry, Jarrod R McClean, and Hartmut Neven. “Quantum simulation of chemistry with sublinear scaling in basis size”. *npj Quantum Information* **5**, 92 (2019).
- [86] Chenyi Zhang, Jiaqi Leng, and Tongyang Li. “Quantum algorithms for escaping from saddle points”. *Quantum* **5**, 529 (2021).
- [87] Yizhou Liu, Weijie J Su, and Tongyang Li. “On quantum speedups for nonconvex optimization via quantum tunneling walks”. *Quantum* **7**, 1030 (2023).
- [88] Stephen Wiesner. “Simulations of many-body quantum systems by a quantum computer” (1996). [arXiv:quant-ph/9603028](https://arxiv.org/abs/quant-ph/9603028).
- [89] Christof Zalka. “Efficient simulation of quantum systems by quantum computers”. *Fortschritte der Physik: Progress of Physics* **46**, 877–879 (1998).
- [90] Ian D Kivlichan, Nathan Wiebe, Ryan Babbush, and Alán Aspuru-Guzik. “Bounding the costs of quantum simulation of many-body physics in real space”. *Journal of Physics A: Mathematical and Theoretical* **50**, 305301 (2017).
- [91] Dong An, Di Fang, and Lin Lin. “Time-dependent unbounded Hamiltonian simulation with vector norm scaling”. *Quantum* **5**, 459 (2021).
- [92] Dong An, Di Fang, and Lin Lin. “Time-dependent Hamiltonian simulation of highly oscillatory dynamics and superconvergence for Schrödinger equation”. *Quantum* **6**, 690 (2022).
- [93] Andrew M Childs, Jiaqi Leng, Tongyang Li, Jin-Peng Liu, and Chenyi Zhang. “Quantum simulation of real-space dynamics”. *Quantum* **6**, 860 (2022).
- [94] Shi Jin, Xiantao Li, and Nana Liu. “Quantum simulation in the semi-classical regime”. *Quantum* **6**, 739 (2022).
- [95] Chia Cheng Chang, Kenneth S. McElvain, Ermal Rrapaj, and Yantao Wu. “Improving Schrödinger equation implementations with gray code for adiabatic quantum computers”. *PRX Quantum* **3**, 020356 (2022).
- [96] Shantanav Chakraborty, Soumyabrata Hazra, Tongyang Li, Changpeng Shao, Xinzhao Wang, and Yuxin Zhang. “Quantum singular value transformation without block encodings: Near-optimal complexity with minimal ancilla” (2025). [arXiv:2504.02385](https://arxiv.org/abs/2504.02385).
- [97] Chandler Davis and William Morton Kahan. “The rotation of eigenvectors by a perturbation. III”. *SIAM Journal on Numerical Analysis* **7**, 1–46 (1970).
- [98] Sergey Bravyi, David P DiVincenzo, and Daniel Loss. “Schrieffer–Wolff transformation for quantum many-body systems”. *Annals of physics* **326**, 2793–2826 (2011).
- [99] Anthony W Knapp. “Basic real analysis”. *Springer Science & Business Media*. (2007).
- [100] Andrew M Childs, Aaron Ostrander, and Yuan Su. “Faster quantum simulation by randomization”. *Quantum* **3**, 182 (2019).

- [101] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Andreas Bengtsson, Sergio Boixo, Michael Broughton, Bob B Buckley, et al. “Observation of separated dynamics of charge and spin in the Fermi-Hubbard model” (2020). [arXiv:2010.07965](#).
- [102] B. Foxen, C. Neill, A. Dunsworth, P. Roushan, B. Chiaro, A. Megrant, J. Kelly, Zijun Chen, K. Satzinger, R. Barends, et al. “Demonstrating a continuous set of two-qubit gates for near-term quantum algorithms”. *Phys. Rev. Lett.* **125**, 120504 (2020).
- [103] Sonika Johri, Shantanu Debnath, Avinash Mocherla, Alexandros Singk, Anupam Prakash, Jungsang Kim, and Iordanis Kerenidis. “Nearest centroid classification on a trapped ion quantum computer”. *npj Quantum Information* **7**, 122 (2021).
- [104] Natansh Mathur, Jonas Landman, Yun Yvonna Li, Martin Strahm, Skander Kazdaghli, Anupam Prakash, and Iordanis Kerenidis. “Medical image classification via quantum neural networks” (2022). [arXiv:2109.01831](#).
- [105] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. “t|ket): a retargetable compiler for NISQ devices”. *Quantum Science and Technology* **6**, 014003 (2020).
- [106] Scott Aaronson and Andris Ambainis. “Quantum search of spatial regions”. In 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings. [Pages 200–209](#). (2003).
- [107] Avatar Tulsi. “Faster quantum-walk algorithm for the two-dimensional spatial search”. *Phys. Rev. A* **78**, 012310 (2008).
- [108] Henning Labuhn, Daniel Barredo, Sylvain Ravets, Sylvain De Léséleuc, Tommaso Macrì, Thierry Lahaye, and Antoine Browaeys. “Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models”. *Nature* **534**, 667–670 (2016).
- [109] Vincent Lienhard, Sylvain de Léséleuc, Daniel Barredo, Thierry Lahaye, Antoine Browaeys, Michael Schuler, Louis-Paul Henry, and Andreas M. Läuchli. “Observing the space- and time-dependent growth of correlations in dynamically tuned synthetic Ising models with antiferromagnetic interactions”. *Phys. Rev. X* **8**, 021070 (2018).
- [110] QuEra Computing Inc. “Bloqade documentation” (2023). [Bloqade.jl](#).

Appendices

A Quantum simulation using Hamiltonian embedding

Notation. Let \mathcal{H} be a Hilbert space, and $A: \mathcal{H} \rightarrow \mathcal{H}$ be a linear operator over \mathcal{H} . Let \mathcal{S} be a subspace of \mathcal{H} , and \mathcal{S}^\perp be the orthogonal complement of \mathcal{S} such that $\mathcal{H} = \mathcal{S} \oplus \mathcal{S}^\perp$. We denote $P_{\mathcal{S}}$ (or $P_{\mathcal{S}^\perp}$) as the projection onto \mathcal{S} (or \mathcal{S}^\perp). We write $A|_{\mathcal{S}} := P_{\mathcal{S}}AP_{\mathcal{S}}$ as the restriction of A in the subspace \mathcal{S} . $\text{Herm}(\mathbb{C}^n)$ represents the space of all n -by- n complex Hermitian matrices.

A.1 Proof of [Theorem 1](#)

Theorem 1 (Hamiltonian simulation with Hamiltonian embedding). *Suppose that H is a (q, η, ϵ) -embedding of A . Then, for a fixed evolution time $t \geq 0$, we have that*

$$\left\| \left(e^{-iHt} \right) \Big|_{\mathcal{S}} - e^{-iAt} \right\| \leq (2\eta\|H\| + \epsilon)t. \quad (\text{A.1})$$

Proof. We define $\mathcal{E}(t) := e^{-iHt} - U^\dagger e^{-iHt}U$, and it follows that

$$\dot{\mathcal{E}}(t) = (-iH)e^{-iHt} - U^\dagger(-iH)e^{-iHt}U = (-iH)\mathcal{E}(t) - i[H - \tilde{H}]U^\dagger e^{-iHt}U, \quad (\text{A.1})$$

where $\tilde{H} = U^\dagger H U$. By the variation-of-parameter formula, we have

$$\mathcal{E}(t) = -i \int_0^t e^{-iH(t-\tau)} [H - \tilde{H}] e^{-i\tilde{H}\tau} d\tau, \quad (\text{A.2})$$

and it follows that

$$\|P_{\mathcal{S}}\mathcal{E}(t)P_{\mathcal{S}}\| \leq \|(H - \tilde{H})P_{\mathcal{S}}\|t = \|[H, U]P_{\mathcal{S}}\|t.$$

We denote $H_0 = P_{\mathcal{S}}\tilde{H}P_{\mathcal{S}}$. By the definition of Hamiltonian embedding, \tilde{H} is block-diagonal and $\|H_0 - A\| \leq \epsilon$. Then, we have

$$\|P_{\mathcal{S}}U^\dagger e^{-iHt}UP_{\mathcal{S}} - e^{-iAt}\| = \|e^{-iH_0t} - e^{-iAt}\| \leq \|H_0 - A\|t \leq \epsilon t.$$

It follows from the triangle inequality that

$$\left\| \left(e^{-iHt} \right) \Big|_{\mathcal{S}} - e^{-iAt} \right\| \leq \|P_{\mathcal{S}}\mathcal{E}(t)P_{\mathcal{S}}\| + \|P_{\mathcal{S}}U^\dagger e^{-iHt}UP_{\mathcal{S}} - e^{-iAt}\| \leq (\|[H, U]P_{\mathcal{S}}\| + \epsilon)t. \quad (\text{A.3})$$

Note that $\|[H, U]\| = \|[H, I] - [H, (I - U)]\| \leq 2\eta\|H\|$, which implies the error bound [\(2.1\)](#). \square

A.2 Rules for building Hamiltonian embeddings

In this section, we introduce several basic rules for building more sophisticated Hamiltonian embeddings from existing ones.

Lemma 4 (Addition). *Let $A_1, A_2 \in \text{Herm}(\mathbb{C}^n)$. For $j = 1, 2$, we suppose that H_j is a (q, η, ϵ_j) -embedding of A_j with unitary transformation U and embedding subspace \mathcal{S} . Then, $H = H_1 + H_2$ is a $(q, \eta, \epsilon_1 + \epsilon_2)$ -embedding of $A = A_1 + A_2$ with the same unitary transformation U and embedding subspace \mathcal{S} .*

Proof. Since the unitary U block-diagonalizes both H_1 and H_2 , it also block-diagonalizes the $H = H_1 + H_2$. Moreover, by the triangle inequality, $\|(U^\dagger H U)_\mathcal{S} - (A_1 + A_2)\| \leq \|(U^\dagger H_1 U)_\mathcal{S} - A_1\| + \|(U^\dagger H_2 U)_\mathcal{S} - A_2\| \leq \epsilon_1 + \epsilon_2$. \square

Lemma 5 (Multiplication). *Let H be a (q, η, ϵ) -embedding of A with unitary transformation U and embedding subspace \mathcal{S} . Then, for any real number α , αH is a $(q, \eta, |\alpha|\epsilon)$ -embedding of αA with the same unitary transformation U and embedding subspace \mathcal{S} .*

Proof. Due to the linearity of matrix multiplication, we have that

$$\|(U^\dagger \alpha H U)_\mathcal{S} - \alpha A\| = |\alpha| \|(U^\dagger H U)_\mathcal{S} - A\| \leq |\alpha| \epsilon.$$

\square

Lemma 6 (Composition). *For $j = 1, 2$, let H_j be a $(q_j, \eta_j, \epsilon_j)$ -embedding of A_j with unitary transformation U_j and embedding subspace \mathcal{S}_j , respectively. Then, $H = H_1 \otimes I + I \otimes H_2$ is a $(q_1 + q_2, \eta_1 + \eta_2, \epsilon_1 + \epsilon_2)$ -embedding of $A = A_1 \otimes I + I \otimes A_2$ with unitary transformation $U = U_1 \otimes U_2$ and embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$.*

Proof. In the full Hilbert space $\mathcal{H}' = \mathbb{C}^{2^{q_1+q_2}}$, we have the embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$, so its orthogonal complement is

$$\mathcal{S}^\perp = (\mathcal{S}_1^\perp \otimes \mathcal{S}_2) \oplus (\mathcal{S}_1 \otimes \mathcal{S}_2^\perp) \oplus (\mathcal{S}_1^\perp \otimes \mathcal{S}_2^\perp).$$

First, we check the rotated Hamiltonian $\tilde{H} := U^\dagger H U$ is block-diagonal in \mathcal{S} and \mathcal{S}^\perp . We observe that $P_{\mathcal{S}^\perp} = P_{\mathcal{S}_1^\perp \otimes \mathcal{S}_2} + P_{\mathcal{S}_1 \otimes \mathcal{S}_2^\perp} + P_{\mathcal{S}_1^\perp \otimes \mathcal{S}_2^\perp}$. It is readily verified that

$$P_{\mathcal{S}_1^\perp \otimes \mathcal{S}_2} (U^\dagger H U) P_{\mathcal{S}_1 \otimes \mathcal{S}_2} = (P_{\mathcal{S}_1^\perp} \tilde{H}_1 P_{\mathcal{S}_1}) \otimes I + (P_{\mathcal{S}_1^\perp} P_{\mathcal{S}_1}) \otimes (P_{\mathcal{S}_2} \tilde{H}_2 P_{\mathcal{S}_2}) = 0.$$

Similarly, we can show that

$$P_{\mathcal{S}_1 \otimes \mathcal{S}_2^\perp} (U^\dagger H U) P_{\mathcal{S}_1 \otimes \mathcal{S}_2} = 0, \quad P_{\mathcal{S}_1^\perp \otimes \mathcal{S}_2^\perp} (U^\dagger H U) P_{\mathcal{S}_1 \otimes \mathcal{S}_2} = 0,$$

which implies $P_{\mathcal{S}^\perp} \tilde{H} P_{\mathcal{S}} = 0$, i.e., \tilde{H} is block-diagonal. Also, for the unitary $U = U_1 \otimes U_2$, we have

$$\|I - U_1 \otimes U_2\| \leq \|(I - U_1) \otimes I\| + \|U_1 \otimes (I - U_2)\| \leq \eta_1 + \eta_2.$$

Finally, we check the approximation error,

$$\begin{aligned} & \|P_{\mathcal{S}} U_1^\dagger \otimes U_2^\dagger (H_1 \otimes I + I \otimes H_2) U_1 \otimes U_2 P_{\mathcal{S}} - (A_1 \otimes I + I \otimes A_2)\| \\ &= \left\| \left[(U_1^\dagger H_1 U_1)|_{\mathcal{S}_1} - A_1 \right] \otimes I_{\mathcal{S}_2} + I_{\mathcal{S}_1} \otimes \left[(U_2^\dagger H_2 U_2)|_{\mathcal{S}_2} - A_2 \right] \right\| \leq \epsilon_1 + \epsilon_2. \end{aligned}$$

\square

Lemma 7 (Tensor product). *For $j = 1, 2$, let H_j be a $(q_j, \eta_j, \epsilon_j)$ -embedding of A_j with unitary transformation U_j and embedding subspace \mathcal{S}_j , respectively. Then, $H = H_1 \otimes H_2$ is a $(q_1 + q_2, \eta_1 + \eta_2, \|A_1\| \epsilon_2 + \|A_2\| \epsilon_1 + \epsilon_1 \epsilon_2)$ -embedding of $H = A_1 \otimes A_2$ with unitary transformation $U = U_1 \otimes U_2$ and embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$.*

Proof. Similar to Rule 3, we can check that $P_{\mathcal{S}^\perp}(U^\dagger H_1 \otimes H_2 U)P_{\mathcal{S}} = 0$ and $\|I - U_1 \otimes U_2\| \leq \eta_1 + \eta_2$. For $j = 1, 2$, we have that $\|(U_j^\dagger H_j U_j)|_{\mathcal{S}_j}\| \leq \|A_j\| + \epsilon_j$. Using this fact, we can estimate the approximation error:

$$\begin{aligned} & \|P_{\mathcal{S}}U_1^\dagger \otimes U_2^\dagger(H_1 \otimes H_2)U_1 \otimes U_2P_{\mathcal{S}} - (A_1 \otimes A_2)\| = \|(U_1^\dagger H_1 U_1)|_{\mathcal{S}_1} \otimes (U_2^\dagger H_2 U_2)|_{\mathcal{S}_2} - A_1 \otimes A_2\| \\ & \leq \|[A_1 - (U_1^\dagger H_1 U_1)|_{\mathcal{S}_1}] \otimes A_2\| + \|(U_1^\dagger H_1 U_1)|_{\mathcal{S}_1} \otimes [A_2 - (U_2^\dagger H_2 U_2)|_{\mathcal{S}_2}]\| \\ & \leq \|A_2\|\epsilon_1 + \|A_1\|\epsilon_2 + \epsilon_1\epsilon_2. \end{aligned}$$

□

Remark 1. The nonzero entries of a Hamiltonian give rise to the adjacency matrix of a graph (ignoring diagonal entries for simplicity). As a result, the composition and tensor product rules can be visualized as two different notions of graph products. For graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the *Cartesian product* of G_1 and G_2 is the graph $G_1 \square G_2$ with vertex set $V_1 \times V_2$ such that vertices (u_1, v_1) and (u_2, v_2) are adjacent iff either $u_1 = u_2$ and v_1 is adjacent to v_2 , or $v_1 = v_2$ and u_1 is adjacent to u_2 . The *tensor product* of G_1 and G_2 is the graph $G_1 \times G_2$ with vertex set $V_1 \times V_2$ such that vertices (u_1, v_2) and (u_2, v_2) are adjacent iff both u_1 is adjacent to u_2 and v_1 is adjacent to v_2 . From the definitions, it follows that the adjacency matrix of $G_1 \square G_2$ is $A_1 \otimes I + I \otimes A_2$, while the adjacency matrix of $G_1 \times G_2$ is $A_1 \otimes A_2$. Therefore, the composition and tensor product rules can be visualized as taking the graph Cartesian and tensor products, respectively.

A.3 Perturbative Hamiltonian embedding

A.3.1 Schrieffer-Wolff theory

We consider the q -qubit operator given in (2.3), i.e., $H = gH^{\text{pen}} + Q$, where Q and H^{pen} can be expressed in block-matrix forms (w.r.t. subspaces \mathcal{S} and \mathcal{S}^\perp):

$$Q = \begin{bmatrix} A & R^\dagger \\ R & B \end{bmatrix}, \quad H^{\text{pen}} = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix}.$$

Therefore, the operator $H = \begin{bmatrix} A & R^\dagger \\ R & G \end{bmatrix}$, where $G = B + gC$.

We define $\Delta_0 := \lambda_{\min}(B) - \lambda_{\max}(A)$ as the spectral distance between the two diagonal blocks in Q , and $\Delta := \lambda_{\min}(G) - \lambda_{\max}(A)$ being the spectral distance between the two diagonal blocks in H . Note that the spectral gaps Δ_0 and Δ are not necessarily positive. By Weyl's inequality, we have

$$\Delta \geq g\lambda_1 + \Delta_0,$$

where $\lambda_1 > 0$ is the minimum eigenvalue of C (recall that \mathcal{S} is the ground-energy subspace of H^{pen}).

By choosing sufficiently large $g > 0$, we can make $\Delta > 0$ and $\Delta \sim g$. For large Δ such that $\|R\|/\Delta \ll 1$, the off-diagonal part in H can be regarded as a perturbation and we can invoke the Schrieffer-Wolff transformation [97, 98].

Theorem 8 (Schrieffer-Wolff transformation). *Define $\kappa := \|R\|/\Delta$. We assume that $\Delta > 0$ and $\kappa < 1/2$. Then, there exists a unitary transformation U such that $\|I - U\| \leq 2\sqrt{2}\kappa$ and $U^\dagger H U$ is block-diagonal. Moreover, if $\kappa < 1/16$, there exists an absolute constant $c > 0$ such that*

$$\left\| \left(U^\dagger H U \right)_{\mathcal{S}} - A \right\| \leq c\|R\|\kappa. \quad (\text{A.4})$$

In other words, H is a $(q, 2\sqrt{2}\kappa, c\|R\|\kappa)$ -embedding of A if $\kappa < 1/16$.

Proof. We define $H_0 = \begin{bmatrix} A & 0 \\ 0 & G \end{bmatrix}$ and $V = \begin{bmatrix} 0 & R \\ R^\dagger & 0 \end{bmatrix}$, so $H = H_0 + V$. Note that $\|V\| = \|R\|$. When $\kappa = \|V\|/\Delta < 1/2$, there is a unique Schrieffer-Wolff transformation U such that $U(H_0 + V)U^\dagger$ is block-diagonal (see [98] for detailed discussions). By [97, Section 1], there exists an *angle* operator Θ such that $\|I - U\| = 2\|\sin \frac{1}{2}\Theta\|$. Let $\mathcal{T} = U\mathcal{S}$ be the rotated subspace, we also have that $\|P_{\mathcal{S}} - P_{\mathcal{T}}\| = \|\sin \Theta\|$. On the other hand, by [98, Lemma 3.1], we can estimate the distance between \mathcal{S} and \mathcal{T} :

$$\|P_{\mathcal{S}} - P_{\mathcal{T}}\| \leq 2\kappa.$$

Since we have $\kappa < 1/2$, the angle operator is bounded by $\pi/2$ [97]. Note that for any angle $|\theta| < \pi/2$, we have

$$\left| \sin \frac{1}{2}\theta \right| = \sqrt{\frac{1}{2}(1 - \cos \theta)} = \sqrt{\frac{1}{2}(1 - \sqrt{1 - \sin^2 \theta})} \leq \frac{\sqrt{2}}{2} |\sin \theta|,$$

which implies that $\|\sin \frac{1}{2}\Theta\| \leq \frac{\sqrt{2}}{2}\|\sin \Theta\|$. It turns out that

$$\|I - U\| \leq \sqrt{2}\|P_{\mathcal{S}} - P_{\mathcal{T}}\| \leq 2\sqrt{2}\kappa.$$

We define the effective low-energy Hamiltonian $H_{\text{eff}} := P_{\mathcal{S}}U^\dagger H U P_{\mathcal{S}}$. The effective Hamiltonian H_{eff} allows a Taylor series expansion [98, 56]:

$$H_{\text{eff}} = \sum_{j=1}^{\infty} H_{\text{eff},j}. \quad (\text{A.5})$$

We define $H_{\text{eff}}(k) = \sum_{j=1}^k H_{\text{eff},j}$. When $\kappa < 1/16$, the series (A.5) converge absolutely and there exists an absolute constant $c > 0$ such that

$$\|H_{\text{eff}} - H_{\text{eff}}(k)\| \leq c \frac{\|V\|^{k+1}}{\Delta^k} = c\|R\|\kappa^k.$$

Note that $H_{\text{eff}}(1) = A$. Therefore, we prove that $\|H_{\text{eff}} - A\| \leq c\|R\|\kappa$. \square

By Theorem 8, a bigger g implies a more accurate Hamiltonian embedding H . The idea is that a large parameter g suppresses the leakage of the quantum evolution from the embedding subspace \mathcal{S} to its orthogonal complement \mathcal{S}^\perp . Based on this intuition, we refer to g as the *penalty coefficient* and the operator H^{pen} as the *penalty Hamiltonian*.

Remark 2. In a perturbative Hamiltonian embedding with penalty coefficient $g > 0$, the error characterization parameters are $\eta \leq 2\sqrt{2}\kappa \sim \frac{1}{g}$, $\epsilon \leq c\|R\|\kappa \sim \frac{\|R\|}{g}$. By choosing a large enough g , the error parameters η and ϵ can be made arbitrarily small. On the other hand, when H^{pen} is fast-forwardable (which is always the case for embedding schemes discussed in this paper, see Appendix B), there exist quantum algorithms (e.g., continuous qDRIFT, see Appendix A.3.3) that can efficiently simulate the Hamiltonian embedding even for large g . Therefore, for simplicity, we will not explicitly specify η and ϵ when discussing a perturbative Hamiltonian embedding in the rest of this paper.

All the basic rules of constructing Hamiltonian embeddings can be reformulated for perturbative Hamiltonian embedding. The proof is similar to Appendix A.2.

Proposition 9.

1. (Addition) For $j = 1, 2$, suppose $H_j = gH^{\text{pen}} + Q_j$ is an embedding of A_j with the same embedding subspace \mathcal{S} . Then, $H = gH^{\text{pen}} + Q_1 + Q_2$ is an embedding of $A = A_1 + A_2$ with embedding subspace \mathcal{S} .
2. (Multiplication) Suppose $H = gH^{\text{pen}} + Q$ is an embedding of A with embedding subspace \mathcal{S} . Then, for any real number α , $H' = gH^{\text{pen}} + \alpha Q$ is an embedding of $A' = \alpha A$.
3. (Composition) For $j = 1, 2$, suppose that the q_j -qubit operator $H_j = gH_j^{\text{pen}} + Q_j$ is an embedding of A_j with the embedding subspace \mathcal{S}_j being the ground-energy subspace of H_j^{pen} . Then, $H = gH^{\text{pen}} + Q$ with

$$H^{\text{pen}} = H_1^{\text{pen}} \otimes I + I \otimes H_2^{\text{pen}}, \quad Q = Q_1 \otimes I + I \otimes Q_2 \quad (\text{A.6})$$

is an embedding of $A = A_1 \otimes I + I \otimes A_2$ with embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$.

4. (Tensor product) For $j = 1, 2$, suppose that $H_j = gH_j^{\text{pen}} + Q_j$ is an embedding of A_j with the embedding subspace \mathcal{S}_j being the ground-energy subspace of H_j^{pen} . Then, $H = gH^{\text{pen}} + Q_1 \otimes Q_2$ with

$$H^{\text{pen}} = H_1^{\text{pen}} \otimes I + I \otimes H_2^{\text{pen}} \quad (\text{A.7})$$

is an embedding of $H = A_1 \otimes A_2$ with embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$.

A.3.2 Improved simulation error analysis

Theorem 10 (Quantum simulation with perturbative embedding). *Let all the notations be the same as above. We assume $\Delta > 0$ and $\kappa < 1/2$. Then, for any fixed $t > 0$, we have*

$$\|P_{\mathcal{S}} e^{-iHt} P_{\mathcal{S}} - e^{-iAt}\| \leq 4\sqrt{2}\kappa \|R\| t. \quad (\text{A.8})$$

Proof. Let $H_0 = P_{\mathcal{S}} H P_{\mathcal{S}} = \text{diag}(A, 0)$, and we define:

$$\mathcal{A}(t) = P_{\mathcal{S}} e^{-iHt} P_{\mathcal{S}}, \quad \mathcal{B}(t) = P_{\mathcal{S}} e^{-iH_0 t} P_{\mathcal{S}}. \quad (\text{A.9})$$

We also define the $\mathcal{E}(t) := \mathcal{A}(t) - \mathcal{B}(t)$. Note that $\mathcal{E}(0) = 0$. We find that

$$\begin{aligned} \frac{d}{dt} \mathcal{A}(t) &= P_{\mathcal{S}} (-iH) (P_{\mathcal{S}} + P_{\mathcal{S}^\perp}) e^{-iHt} P_{\mathcal{S}} = -iH_0 \mathcal{A}(t) - iR e^{-iHt} P_{\mathcal{S}}, \\ \frac{d}{dt} \mathcal{B}(t) &= P_{\mathcal{S}} (-iH_0) e^{-iH_0 t} P_{\mathcal{S}} = -iH_0 \mathcal{B}(t). \end{aligned}$$

Therefore, we have $\dot{\mathcal{E}}(t) = \dot{\mathcal{A}}(t) - \dot{\mathcal{B}}(t) = -iH_0 \mathcal{E}(t) - iR e^{-iHt} P_{\mathcal{S}}$. By the variation-of-parameter formula [99, Theorem 4.9], we represent the function $\mathcal{E}(t)$ as the time integral:

$$\mathcal{E}(t) = -i \int_0^t \exp(-iH_0(t-\tau)) R \exp(-iH\tau) P_{\mathcal{S}} d\tau, \quad (\text{A.10})$$

so we can estimate the error (recall that $R = P_{\mathcal{S}} H P_{\mathcal{S}^\perp}$):

$$\begin{aligned} \|\mathcal{E}(t)\| &\leq \int_0^t \|P_{\mathcal{S}} H P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| d\tau \leq \int_0^t \|R P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| d\tau \\ &\leq \int_0^t \|R\| (4\sqrt{2}\kappa) d\tau = 4\sqrt{2}\kappa \|R\| t. \end{aligned}$$

In the second step, we use $P_{\mathcal{S}^\perp} = P_{\mathcal{S}^\perp}^2$. In the second to last step, we invoke [Lemma 11](#). \square

Lemma 11. *Assume that $\kappa < 1/2$ and let $\tau \geq 0$ be an arbitrary real number. Then, we have*

$$\|P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| \leq 4\sqrt{2}\kappa. \quad (\text{A.11})$$

Proof. By [Theorem 8](#), there exists a Schrieffer-Wolff transformation U such that

$$U^\dagger H U = \begin{bmatrix} H_{\text{eff}} & 0 \\ 0 & * \end{bmatrix},$$

where $H_{\text{eff}} = P_{\mathcal{S}} U^\dagger H U P_{\mathcal{S}}$. We remark that $P_{\mathcal{S}^\perp} U^\dagger \exp(-iH\tau) U P_{\mathcal{S}} = 0$. To see this, note that U block-diagonalizes H , so does $\exp(-iU H U^\dagger \tau)$:

$$P_{\mathcal{S}^\perp} U \exp(-iH\tau) U^\dagger P_{\mathcal{S}} = P_{\mathcal{S}^\perp} \exp(-iU H U^\dagger \tau) P_{\mathcal{S}} = 0.$$

Now, we bound the cross term $P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}$ by,

$$\begin{aligned} \|P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| &= \|P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}} - P_{\mathcal{S}^\perp} U^\dagger \exp(-iH\tau) U P_{\mathcal{S}}\| \\ &\leq \|\exp(-iH\tau) - U^\dagger \exp(-iH\tau) U\| = \|U \exp(-iH\tau) - \exp(-iH\tau) U\|. \end{aligned}$$

Then, by [Theorem 8](#), we conclude that

$$\|P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| \leq \|(U - I) \exp(-iH\tau) - \exp(-iH\tau)(U - I)\| \leq 2\|U - I\| \leq 4\sqrt{2}\kappa.$$

□

A.3.3 Quantum simulation of perturbative embedding

The perturbative Hamiltonian embedding $H^{\text{ebd}} = gH^{\text{pen}} + Q$ often comes with a large penalty coefficient g . This could lead to significant simulation error if one uses a standard quantum algorithm (e.g., a product formula) to simulate H^{ebd} . In this section, we discuss two types of quantum simulation protocols (product formulas and qDRIFT) for perturbative Hamiltonian embedding. It turns out that qDRIFT is friendly to near-term devices with minimal overheads in terms of g .

First, we consider the standard first-order Trotter formula:

$$e^{-iH^{\text{ebd}}t} \approx U_1(t) := \prod_{k=1}^r e^{-iQt/r} e^{-igH^{\text{pen}}t/r}. \quad (\text{A.12})$$

By [\[34\]](#), the Trotter error scales like

$$\|e^{-iH^{\text{ebd}}t} - U_1(t)\| \leq O\left(\frac{g[H^{\text{pen}}, Q]t^2}{r}\right). \quad (\text{A.13})$$

To simulate the embedding Hamiltonian up to an additive error ϵ , we need to choose the Trotter number

$$r = O\left(\frac{g[H^{\text{pen}}, Q]t^2}{\epsilon}\right). \quad (\text{A.14})$$

Similarly, the second-order Trotter-Suzuki formula requires the Trotter number

$$r = O\left(\frac{\Gamma t^{3/2}}{\epsilon^{1/2}}\right), \quad (\text{A.15})$$

where $\Gamma = \left(\frac{g}{12} \|[Q, [Q, H^{\text{pen}}]]\| + \frac{g^2}{24} \|[H^{\text{pen}}, [H^{\text{pen}}, Q]]\| \right)^{1/2}$.

We can improve the first-order Trotter formula by randomization [100], which turns out to be effectively a second-order product formula:

$$\|e^{-iH^{\text{ebd}}t} - U_{1,\text{rand}}(t)\| \leq O\left(\frac{\Lambda^3 t^3}{r^2}\right), \quad (\text{A.16})$$

where $U_{1,\text{rand}}(t)$ is the randomized first-order Trotter formula, and $\Lambda = \max\{g\|H^{\text{pen}}\|, \|Q\|\}$. It follows that we need the Trotter number

$$r = O\left(\frac{\Lambda^{3/2} t^{3/2}}{\epsilon^{1/2}}\right) \quad (\text{A.17})$$

in order to achieve an ϵ -accurate simulation of H' . A second-order (deterministic) Trotter formula will require that $r = O(gt^{3/2}\epsilon^{-1/2})$, but potentially have a better pre-factor in terms of the commutators of H^{pen} and Q [34].

A major drawback of the aforementioned quantum simulation protocols we discussed above is that their simulation error has explicit dependence on the penalty coefficient g , which could be very huge in practice. To address this issue, we consider quantum simulation in the interaction picture. For a Hamiltonian $H = A + B$, we transform it to the interaction picture:

$$H_I(t) = e^{iAt} B e^{-iAt}, \quad (\text{A.18})$$

and consider the Schrödinger equation

$$i \frac{d}{dt} |\psi_I(t)\rangle = H_I(t) |\psi_I(t)\rangle, \quad (\text{A.19})$$

subject to $|\psi_I(0)\rangle = |\psi_0\rangle$. One can easily verify that the state vector $|\psi(t)\rangle = e^{-iHt} |\psi_0\rangle$ can be recovered by

$$|\psi(t)\rangle = e^{-iAt} |\psi_I(t)\rangle. \quad (\text{A.20})$$

Based on this observation, digital quantum simulation protocols for time-dependent Hamiltonian that has L^1 -norm scaling can be used to simulate $H_I(t)$ to achieve an error scaling only depends on $\|B\|$, given that A can be fast-forwardable.

In our construction of Hamiltonian embedding, H^{pen} is often fast-forwardable, so we can leverage the interaction-picture quantum simulation protocols to simulate

$$H_I^{\text{ebd}}(t) = e^{igH^{\text{pen}}t} Q e^{-igH^{\text{pen}}t} \quad (\text{A.21})$$

and achieve g -independent error scaling. There are a handful of quantum algorithms that can achieve l^1 -norm scaling for interaction picture Hamiltonian, e.g., truncated Dyson series [18, 55], continuous qDRIFT [55], qHOP [92], etc. Unfortunately, the truncated Dyson series method and the qHOP method require complicated control flows and advanced input models (e.g., block-encodings) that are infeasible for near-term quantum computers. On the other hand, continuous qDRIFT relies on a simple product formula and a classical stochastic sampling protocol, which is possible to implement on near-term devices. The details of qDRIFT are presented in Algorithm 1.

We can prove that when M, K are large enough,

$$\langle \psi_0 | e^{iH^{\text{ebd}}T} O e^{-iH^{\text{ebd}}T} | \psi_0 \rangle \approx \tilde{E}. \quad (\text{A.22})$$

Algorithm 1: Quantum simulation by qDRIFT.

input : An initial state $|\psi_0\rangle$, number of samples M , Trotter number K , total evolution time T .

output: An estimate of the measurement result \tilde{E} .

```
1 Let  $\Delta t = T/K$ ;  
2 for  $j \leftarrow 1$  to  $M$  do  
3   for  $k \leftarrow 1$  to  $K$  do  
4     Sample a uniformly random number  $\xi_k \in [(k-1)\Delta t, k\Delta t]$ ;  
5     Propagate the quantum state  
6      $|\psi_k\rangle = e^{-iH_I^{\text{ebd}}(\xi_k)\Delta t} |\psi_{k-1}\rangle = e^{igH^{\text{pen}}\xi_k} e^{-iQ\Delta t} e^{-igH^{\text{pen}}\xi_k} |\psi_{k-1}\rangle$ .  
7   Measure the quantum state  $|\psi_K\rangle$  with the observable  
    $M_I = e^{igH^{\text{pen}}T} O e^{-igH^{\text{pen}}T}$  and compute  $m_j = \langle \psi_K | M_I | \psi_K \rangle$ ;  
8 Compute the sample mean  $\tilde{E} = \frac{1}{M} \sum_{j=1}^M m_j$ .
```

More precisely, by [55], the Trotter number required by continuous qDRIFT to simulate the interaction-picture Hamiltonian $H'_I(t)$ is

$$r = O\left(\frac{\|Q\|^2 t^2}{\epsilon}\right). \quad (\text{A.23})$$

Notably, the error bound in (A.23) is independent of the penalty coefficient g , which allows us to simulate the embedding Hamiltonian H^{ebd} with large g .

B Hamiltonian embedding of sparse matrices

In this section, we construct several Hamiltonian embeddings whose locality has explicit dependence on the sparsity structure of the target matrix. Such embeddings are advantageous for special sparse matrices, e.g., tridiagonal matrices.

B.1 Band matrix

A band matrix is a sparse matrix whose non-zero entries are confined to a diagonal band. We say a matrix A has bandwidth d if $A_{i,j} = 0$ for any $i, j = 1, \dots, n$ such that $|i - j| > d$. For example, a tri-diagonal matrix is of bandwidth 1. Clearly, a band matrix with bandwidth d is $(2d + 1)$ -sparse.

In this section, we discuss the Hamiltonian embedding of Hermitian band matrices. We will introduce two types of embeddings whose locality explicitly depends on the bandwidth of a target matrix.

B.1.1 Unary embedding

Definition 2 (Unary code). Given an integer $j \in \{1, \dots, n\}$, the unary coding of j , denoted by u_j , is the $(n - 1)$ -bit string whose first⁷ $(j - 1)$ bits are 1's and the last $(n - j)$ bits are 0's.

⁷In this paper, we use little-endian ordering, i.e., we enumerate the bit of a string *from right to left*. For example, in the string 100, we refer the first bit to be 0, and the last bit to be 1.

In [Table 7](#), we list the unary code for integers $1 \leq j \leq 8$.

Unary code ($n = 8$)			
Basis index	Codeword	Basis index	Codeword
1	0000000	5	0001111
2	0000001	6	0011111
3	0000011	7	0111111
4	0000111	8	1111111

Table 7: 7-bit unary code representing integers $\{1, 2, \dots, 8\}$.

The unary embedding of an n -by- n Hermitian matrix A is an $(n-1)$ -qubit quantum operator with the embedding subspace spanned by the unary code, i.e., $\mathcal{S} = \text{span}\{|u_1\rangle, \dots, |u_n\rangle\}$. It is easily verified that \mathcal{S} is the ground-energy subspace of the following quantum operator:

$$H_{\text{unary}}^{\text{pen}} = - \sum_{j=1}^{n-2} Z_{j+1} Z_j + Z_1 - Z_{n-1}. \quad (\text{B.1})$$

Example 1. For $n = 4$, the penalty Hamiltonian $H_{\text{unary}}^{\text{pen}}$ has 2 degenerate eigenspaces:

$$E_0 = -2 : |000\rangle, |001\rangle, |011\rangle, |111\rangle, \quad (\text{B.2})$$

$$E_1 = 2 : |010\rangle, |100\rangle, |101\rangle, |110\rangle, \quad (\text{B.3})$$

and the energy gap between the two eigenspaces is 4. In fact, for any $n \geq 3$, the spectral gap between the ground and the first-excited energy levels of $H_{\text{unary}}^{\text{pen}}$ is always 4.

For $1 \leq j < k \leq n$, we find that

$$\langle u_j | (X_{k-1} \otimes \dots \otimes X_{j+1} \otimes X_j) | u_k \rangle = 1, \quad \langle u_k | (X_{k-1} \otimes \dots \otimes X_{j+1} \otimes X_j) | u_j \rangle = 1, \quad (\text{B.4})$$

$$\langle u_j | (X_{k-1} \otimes \dots \otimes X_{j+1} \otimes Y_j) | u_k \rangle = -i, \quad \langle u_k | (X_{k-1} \otimes \dots \otimes X_{j+1} \otimes Y_j) | u_j \rangle = i. \quad (\text{B.5})$$

We denote \hat{n}_j as the number operator at site j :

$$\hat{n}_j := \frac{1}{2} (I - Z_j). \quad (\text{B.6})$$

Proposition 12 (Unary embedding). *Given an n -dimensional Hermitian matrix A , we define the following $(n-1)$ -qubit quantum operator:*

$$Q_A = \left(\alpha_1 I + \sum_{2 \leq j \leq n} (\alpha_j - \alpha_{j-1}) \hat{n}_{j-1} \right) + \sum_{1 \leq j < k \leq n} X_{k-1} \otimes \dots \otimes X_{j+1} \otimes (\alpha_{j,k} X_j - \beta_{j,k} Y_j), \quad (\text{B.7})$$

where α_j is the j -th diagonal entry of A , $\alpha_{j,k}$ and $\beta_{j,k}$ are the real and imaginary part of the (j, k) -th entry of A . Then, the quantum operator

$$H_A^{\text{ebd}} = g H_{\text{unary}}^{\text{pen}} + Q_A \quad (\text{B.8})$$

is an embedding of A . Moreover, the embedding Hamiltonian H_A^{ebd} is $\max(d, 2)$ -local if A has band width d .

Proof. For any $k = 1, \dots, n$,

$$\langle u_k | \left(\alpha_1 I + \sum_{2 \leq j \leq n} (\alpha_j - \alpha_{j-1}) \hat{n}_{j-1} \right) | u_k \rangle = \alpha_1 + \sum_{2 \leq j \leq k} (\alpha_j - \alpha_{j-1}) = \alpha_k. \quad (\text{B.9})$$

For any $\ell, m \in \{1, \dots, n\}$,

$$\langle u_\ell | Q_A | u_m \rangle = \langle u_\ell | \left(\sum_{1 \leq j < k \leq n} X_{k-1} \otimes \cdots \otimes X_{j+1} \otimes (\alpha_{j,k} X_j - \beta_{j,k} Y_j) \right) | u_m \rangle \quad (\text{B.10})$$

$$= \langle u_\ell | (X_{m-1} \otimes \cdots \otimes X_{\ell+1} \otimes (\alpha_{\ell,m} X_\ell - \beta_{\ell,m} Y_\ell)) | u_m \rangle \quad (\text{B.11})$$

$$= \alpha_{\ell,m} + \beta_{\ell,m} i. \quad (\text{B.12})$$

Letting $P_S = \sum_{k=1}^n |u_k\rangle\langle u_k|$, it follows that $P_S Q_A P_S = A$ and thus H_A^{ebd} is an embedding of A . If A has bandwidth d , then $\alpha_{j,k} = \beta_{j,k} = 0$ for $|j - k| > d$, making Q_A d -local. Since $H_{\text{unary}}^{\text{pen}}$ is 2-local, H_A^{ebd} is $\max(d, 2)$ -local. \square

B.1.2 Antiferromagnetic embedding

Definition 3 (Antiferromagnetic code). Given an integer $j \in \{1, \dots, n\}$, the antiferromagnetic coding of j , denoted by a_j , is an $(n - 1)$ -bit string defined as follows:

1. when $j = 1$, a_j is the 0-1 alternating string starting with 0;
2. when $j = 2, \dots, n$, a_j is generated by flipping the $(j - 1)$ -th bit in the previous codeword a_{j-1} .

In [Table 8](#), we list the antiferromagnetic code for integers $1 \leq j \leq 8$.

Antiferromagnetic code ($n = 8$)			
Basis index	Codeword	Basis index	Codeword
1	0101010	5	0100101
2	0101011	6	0110101
3	0101001	7	0010101
4	0101101	8	1010101

Table 8: 7-bit antiferromagnetic code representing integers $\{1, 2, \dots, 8\}$.

The antiferromagnetic embedding of an n -by- n Hermitian matrix A is an $(n - 1)$ -qubit quantum operator with the embedding subspace spanned by the antiferromagnetic code, i.e., $\mathcal{S} = \text{span}\{|a_1\rangle, \dots, |a_n\rangle\}$. It is easily verified that \mathcal{S} is the ground-energy subspace of the following quantum operator:

$$H_{\text{antiferro}}^{\text{pen}} = \sum_{j=1}^{n-2} Z_{j+1} Z_j + Z_1 + (-1)^{n-1} Z_{n-1}. \quad (\text{B.13})$$

Example 2. For $n = 4$, the penalty Hamiltonian $H_{\text{antiferro}}^{\text{pen}}$ has 2 degenerate eigenspaces:

$$E_0 = -2 : |001\rangle, |010\rangle, |011\rangle, |101\rangle, \quad (\text{B.14})$$

$$E_1 = 2 : |000\rangle, |100\rangle, |110\rangle, |111\rangle, \quad (\text{B.15})$$

and the energy gap between the two eigenspaces is 4. In fact, for any $n \geq 3$, the spectral gap between the ground and the first-excited energy levels of $H_{\text{antiferro}}^{\text{pen}}$ is always 4.

For any $1 \leq j < k \leq n$, we find that

$$\langle a_j | (X_{k-1} \otimes \cdots \otimes X_{j+1} \otimes X_j) | a_k \rangle = 1, \quad \langle a_k | (X_{k-1} \otimes \cdots \otimes X_{j+1} \otimes X_j) | a_j \rangle = 1, \quad (\text{B.16})$$

$$\langle a_j | (X_{k-1} \otimes \cdots \otimes X_{j+1} \otimes Y_j) | a_k \rangle = -i, \quad \langle a_k | (X_{k-1} \otimes \cdots \otimes X_{j+1} \otimes Y_j) | a_j \rangle = i. \quad (\text{B.17})$$

Proposition 13 (Antiferromagnetic embedding). *Given an n -dimensional Hermitian matrix A , we define the following $(n-1)$ -qubit quantum operator:*

$$Q_A = \left(\gamma I + \sum_{2 \leq j \leq n} (-1)^j (\alpha_j - \alpha_{j-1}) \hat{n}_{j-1} \right) + \sum_{1 \leq j < k \leq n} X_{k-1} \otimes \cdots \otimes X_{j+1} \otimes (\alpha_{j,k} X_j - \beta_{j,k} Y_j), \quad (\text{B.18})$$

where α_j is the j -th diagonal entry of A , $\alpha_{j,k}$ and $\beta_{j,k}$ are the real and imaginary part of the (j,k) -th entry of A , and

$$\gamma := \sum_{j=1}^n (-1)^{j+1} \alpha_j. \quad (\text{B.19})$$

Then, the quantum operator

$$H_A^{\text{ebd}} = g H_{\text{antiferro}}^{\text{pen}} + Q_A \quad (\text{B.20})$$

is an embedding of A . Moreover, the embedding Hamiltonian H_A^{ebd} is $\max(d, 2)$ -local if A has band width d .

Proof. Note that

$$\left(\gamma I + \sum_{2 \leq j \leq n} (-1)^j (\alpha_j - \alpha_{j-1}) \hat{n}_{j-1} \right) = \alpha_1 + \sum_{2 \leq j \leq n} (\alpha_j - \alpha_{j-1}) \left(\frac{I - (-1)^j Z_{j-1}}{2} \right). \quad (\text{B.21})$$

Then for any $k = 1, \dots, n$,

$$\langle a_k | \left(\alpha_1 + \sum_{2 \leq j \leq n} (\alpha_j - \alpha_{j-1}) \left(\frac{I - (-1)^j Z_{j-1}}{2} \right) \right) | a_k \rangle = \alpha_1 + \sum_{2 \leq j \leq k} (\alpha_j - \alpha_{j-1}) = \alpha_k. \quad (\text{B.22})$$

For $1 \leq \ell < m \leq n$,

$$\langle a_\ell | Q_A | a_m \rangle = \langle a_\ell | \left(\sum_{1 \leq j < k \leq n} X_{k-1} \otimes \cdots \otimes X_{j+1} \otimes (\alpha_{j,k} X_j - \beta_{j,k} Y_j) \right) | a_m \rangle \quad (\text{B.23})$$

$$= \langle a_\ell | (X_{m-1} \otimes \cdots \otimes X_{\ell+1} \otimes (\alpha_{\ell,m} X_\ell - \beta_{\ell,m} Y_\ell)) | a_m \rangle \quad (\text{B.24})$$

$$= \alpha_{\ell,m} + \beta_{\ell,m} i. \quad (\text{B.25})$$

Letting $P_S = \sum_{k=1}^n |a_k\rangle\langle a_k|$, it follows that $P_S Q_A P_S = A$ and thus H_A^{ebd} is an embedding of A . If A has bandwidth d , then $\alpha_{j,k} = \beta_{j,k} = 0$ for $|j-k| > d$, making Q_A d -local. Since $H_{\text{unary}}^{\text{pen}}$ is 2-local, H_A^{ebd} is $\max(d, 2)$ -local. \square

B.2 Banded circulant matrix

In this section, we discuss how to embed (real) symmetric circulant matrices of bandwidth 1 (i.e., tridiagonal) into 2-local quantum operators. Our construction can be readily generalized to circulant matrices of bandwidth d . It is worth noting that the embedding discussed in this section only applies to circulant matrices of even dimensions.

Given an even number n , we consider the following n -by- n circulant matrix:

$$C = \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & 0 & 1 \\ 1 & \dots & 0 & 1 & 0 \end{bmatrix}. \quad (\text{B.26})$$

The matrix C is precisely the adjacency matrix of the (unweighted) n -node cycle graph. We introduce two types of embeddings for the matrix C .

B.2.1 Circulant unary embedding

Definition 4 (Circulant unary code). Given an integer $j \in \{1, \dots, n\}$, the circulant unary coding of j , denoted by c_j , is an $\frac{n}{2}$ -bit string defined as follows:

1. when $j = 1, \dots, n/2$, the first $(j-1)$ bits of c_j are 1's and the remaining $(n/2+1-j)$ bits are 0's;
2. when $j = n/2 + 1, \dots, n$, the codeword c_j is the negation of $c_{j-n/2}$.

In [Table 9](#), we list the circulant unary code for integers $1 \leq j \leq 8$.

Circulant unary code ($n = 8$)			
Basis index	Codeword	Basis index	Codeword
1	0000	5	1111
2	0001	6	1110
3	0011	7	1100
4	0111	8	1000

Table 9: 4-bit circulant unary code representing integers $\{1, 2, \dots, 8\}$.

The circulant unary embedding of the matrix C is an $\frac{n}{2}$ -qubit quantum operator with the embedding subspace spanned by the circulant unary code, i.e., $\mathcal{S} = \text{span}\{|c_1\rangle, \dots, |c_n\rangle\}$. It is easily verified that \mathcal{S} is the ground-energy subspace of the following quantum operator:

$$H_{\text{circ-unary}}^{\text{pen}} = - \sum_{j=1}^{n/2-1} Z_{j+1}Z_j + Z_{n/2}Z_1. \quad (\text{B.27})$$

Example 3. For $n = 8$, the penalty Hamiltonian $H_{\text{circ-unary}}^{\text{pen}}$ has 2 degenerate eigenspaces:

$$E_0 = -2 : |0000\rangle, |0001\rangle, |0011\rangle, |0111\rangle, |1000\rangle, |1100\rangle, |1110\rangle, |1111\rangle, \quad (\text{B.28})$$

$$E_1 = 2 : |0010\rangle, |0100\rangle, |0101\rangle, |0110\rangle, |1001\rangle, |1010\rangle, |1011\rangle, |1101\rangle, \quad (\text{B.29})$$

and the energy gap between the two eigenspaces is 4. In fact, for any $n \geq 4$, the spectral gap between the ground and the first-excited energy levels of $H_{\text{circ-unary}}^{\text{pen}}$ is always 4.

Proposition 14 (Circulant unary embedding). *We define the following $\frac{n}{2}$ -qubit quantum operator:*

$$Q_C = \sum_{j=1}^{n/2} X_j. \quad (\text{B.30})$$

Then, the quantum operator

$$H_C^{\text{ebd}} = gH_{\text{circ-unary}}^{\text{pen}} + Q_C \quad (\text{B.31})$$

is an embedding of the circulant matrix C as in (B.26).

Proof. Observe that Q_C is the adjacency matrix of the $(n/2)$ -dimensional hypercube, containing edges between bitstrings with Hamming distance 1. Therefore, for any $1 \leq k < \ell \leq n$,

$$\langle c_k | Q_C | c_\ell \rangle = \begin{cases} 1 & \text{if } |k - \ell| = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{B.32})$$

□

B.2.2 Circulant antiferromagnetic embedding

Definition 5 (Circulant antiferromagnetic code). Given an integer $j \in \{1, \dots, n\}$, the circulant antiferromagnetic coding of j , denoted by f_j , is an $\frac{n}{2}$ -bit string defined as follows:

1. when $j = 1$, the first codeword f_1 is the 0-1 alternating string starting with 0;
2. when $j = 2, \dots, n/2$, f_j is generated by flipping the $(j - 1)$ -th bit in the previous codeword f_{j-1} ;
3. when $j = n/2 + 1, \dots, n$, the codeword f_j is the negation of $f_{j-n/2}$.

In Table 10, we list the circulant antiferromagnetic code for integers $1 \leq j \leq 8$.

Circulant antiferromagnetic code ($n = 8$)			
Basis index	Codeword	Basis index	Codeword
1	1010	5	0101
2	1011	6	0100
3	1001	7	0110
4	1101	8	0010

Table 10: 4-bit circulant antiferromagnetic code representing integers $\{1, 2, \dots, 8\}$.

The circulant antiferromagnetic embedding of the matrix C is an $\frac{n}{2}$ -qubit quantum operator with the embedding subspace spanned by the circulant antiferromagnetic code, i.e., $\mathcal{S} = \{|f_1\rangle, \dots, |f_n\rangle\}$. It is easily verified that \mathcal{S} is the ground-energy subspace of the following quantum operator:

$$H_{\text{circ-antiferro}}^{\text{pen}} = \sum_{j=1}^{n/2-1} Z_{j+1}Z_j - (-1)^{n/2}Z_{n/2}Z_1. \quad (\text{B.33})$$

Example 4. For $n = 8$, the penalty Hamiltonian $H_{\text{circ-antiferro}}^{\text{pen}}$ has 2 degenerate eigenspaces:

$$E_0 = -2 : |0010\rangle, |0100\rangle, |0101\rangle, |0110\rangle, |1001\rangle, |1010\rangle, |1011\rangle, |1101\rangle, \quad (\text{B.34})$$

$$E_1 = 2 : |0000\rangle, |0001\rangle, |0011\rangle, |0111\rangle, |1000\rangle, |1100\rangle, |1110\rangle, |1111\rangle, \quad (\text{B.35})$$

and the energy gap between the two eigenspaces is 4. In fact, for any $n \geq 4$, the spectral gap between the ground and the first-excited energy levels of $H_{\text{circ-antiferro}}^{\text{pen}}$ is always 4.

Proposition 15 (Circulant antiferromagnetic embedding). *We define the following $\frac{n}{2}$ -qubit quantum operator:*

$$Q_C = \sum_{j=1}^{n/2} X_j. \quad (\text{B.36})$$

Then, the quantum operator

$$H_C^{\text{ebd}} = gH_{\text{circ-antiferro}}^{\text{pen}} + Q_C \quad (\text{B.37})$$

is an embedding of the circulant matrix C as in (B.26).

Proof. Observe that Q_C is the adjacency matrix of the $(n/2)$ -dimensional hypercube, containing edges between bitstrings with Hamming distance 1. Therefore, for any $1 \leq k < \ell \leq n$,

$$\langle f_k | Q_C | f_\ell \rangle = \begin{cases} 1 & \text{if } |k - \ell| = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{B.38})$$

□

B.3 Arbitrary sparse matrices: one-hot embedding

For arbitrary Hermitian matrix H , the bandwidth of H could be huge even if it is sparse. For such target matrices, the unary or antiferromagnetic embedding protocol will lead to a Hamiltonian embedding with a high locality (depending on the bandwidth of H). In this section, we introduce a new type of Hamiltonian embedding based on the one-hot code that has a constant locality.

Definition 6 (One-hot code). Given an integer $j \in \{1, \dots, n\}$, the one-hot coding of j , denoted by h_j , is an n -bit string whose all but the j -th bit are 0's.

In Table 11, we list the one-hot code for integers $1 \leq j \leq 8$.

One-hot code ($n = 8$)			
Basis index	Codeword	Basis index	Codeword
1	00000001	5	00010000
2	00000010	6	00100000
3	00000100	7	01000000
4	00001000	8	10000000

Table 11: 8-bit antiferromagnetic code representing integers $\{1, 2, \dots, 8\}$.

B.3.1 One-hot embedding with penalty

We remark that the edit distance between any two codewords in the one-hot code is 2. This nice property allows us to embed any matrix element with a 2-local operator. Indeed, for $1 \leq j < k \leq n$, we find that

$$\langle h_j | (X_k \otimes X_j) | h_k \rangle = 1, \quad \langle h_k | (X_k \otimes X_j) | h_j \rangle = 1, \quad (\text{B.39})$$

$$\langle h_j | (X_k \otimes Y_j) | h_k \rangle = i, \quad \langle h_k | (X_k \otimes Y_j) | h_j \rangle = -i. \quad (\text{B.40})$$

Meanwhile, the matrix elements on the main diagonal of the target Hamiltonian can be embedded into the number operator. For $1 \leq j, k \leq n$, we observe that (note that the number operator \hat{n}_j is defined in (B.6))

$$\langle h_j | \hat{n}_j | h_j \rangle = \begin{cases} 0 & (j \neq k) \\ 1 & (j = k). \end{cases} \quad (\text{B.41})$$

Denote \mathcal{S} as the embedding subspace spanned by the one-hot code states. Unfortunately, neither $X_k \otimes X_j$ nor $X_k \otimes Y_j$ is block-diagonalized by \mathcal{S} and \mathcal{S}^\perp . Therefore, we need a penalty Hamiltonian in the full embedding Hamiltonian.

The one-hot code are n -bit strings with Hamming weight 1, so they span the $(n-2)$ -eigenspace of the ‘‘sum-of- Z ’’ operator:

$$H_{\text{one-hot}}^{\text{pen}} = \sum_{j=1}^n Z_j. \quad (\text{B.42})$$

We also note that the one-hot code spans the ground-energy subspace of the following operator:

$$H_{\text{one-hot,g}}^{\text{pen}} = \left(\sum_{j=1}^n \hat{n}_j - 1 \right)^2. \quad (\text{B.43})$$

Example 5. For $n = 3$, the penalty Hamiltonian $H_{\text{one-hot}}^{\text{pen}}$ has 4 eigenspaces that correspond to computational basis with Hamming weight 3, 2, 1, 0, respectively:

$$E_0 = -3 : |111\rangle, \quad (\text{B.44})$$

$$E_1 = -1 : |011\rangle, |101\rangle, |110\rangle, \quad (\text{B.45})$$

$$E_2 = 1 : |001\rangle, |010\rangle, |100\rangle, \quad (\text{B.46})$$

$$E_3 = 3 : |000\rangle. \quad (\text{B.47})$$

Clearly, the one-hot code spans the 1-eigenspace. Meanwhile, the penalty Hamiltonian $H_{\text{one-hot,g}}^{\text{pen}}$ has 3 eigenspaces:

$$E_0 = 0 : |011\rangle, |101\rangle, |110\rangle, \quad (\text{B.48})$$

$$E_1 = 1 : |000\rangle, |011\rangle, |101\rangle, |110\rangle, \quad (\text{B.49})$$

$$E_2 = 4 : |111\rangle, \quad (\text{B.50})$$

$$(\text{B.51})$$

and the one-hot code spans the ground-energy subspace.

Proposition 16 (One-hot embedding with penalty). *Given an n -dimensional Hermitian matrix A , we define the following n -qubit quantum operator:*

$$Q_A = \left(\sum_{j=1}^n \alpha_j \hat{n}_j \right) + \left(\sum_{1 \leq j < k \leq n} \alpha_{j,k} X_k \otimes X_j + \beta_{j,k} X_k \otimes Y_j \right), \quad (\text{B.52})$$

where α_j is the j -th diagonal entry of A , $\alpha_{j,k}$ and $\beta_{j,k}$ are the real and imaginary part of the (j,k) -th entry of A , respectively. Then, the 2-local quantum operator

$$H_A^{\text{ebd}} = g H_{\text{one-hot}}^{\text{pen}} + Q_A \quad (\text{B.53})$$

is an embedding of A .

Proof. For any $j = 1, \dots, n$,

$$\langle h_j | H_A^{\text{ebd}} | h_j \rangle = \langle h_j | \left(\sum_{j=1}^n \alpha_j \hat{n}_j \right) | h_j \rangle = \alpha_j. \quad (\text{B.54})$$

For $1 \leq \ell < m \leq n$,

$$\langle h_\ell | H_A^{\text{ebd}} | h_m \rangle = \frac{1}{2} \langle h_\ell | \left(\sum_{1 \leq j < k \leq n} \alpha_{j,k} (X_k \otimes X_j) + \beta_{j,k} (X_k \otimes Y_j) \right) | h_m \rangle \quad (\text{B.55})$$

$$= \langle h_\ell | (\alpha_{\ell,m} (X_m \otimes X_\ell) + \beta_{\ell,m} (X_m \otimes Y_\ell)) | h_m \rangle \quad (\text{B.56})$$

$$= \alpha_{\ell,m} + \beta_{\ell,m} i \quad (\text{B.57})$$

Thus, H_A^{ebd} is an embedding of A . \square

B.3.2 Penalty-free one-hot embedding

The one-hot code allows one to design the perturbation Hamiltonian to be block-diagonal, thereby eliminating the need for a penalty term. In physics, the one-hot code is known as the single-excitation subspace, and the leakage can be fully eliminated by considering excitation-preserving Hamiltonians. In the digital setting, this corresponds to excitation-preserving gates, which often arise in simulations of fermionic systems [101].

There are two parameterized gates which are particularly relevant to the one-hot code. The first is the fSim gate [102] which acts as a Pauli- X rotation in the $\{|01\rangle, |10\rangle\}$ subspace. The other is the RBS gate [103, 104], which acts as a Pauli- Y rotation in the $\{|01\rangle, |10\rangle\}$ subspace. For both of these gates, we can ignore the local phase in the $\{|00\rangle, |11\rangle\}$ subspace. Based on these observations, we consider the following two Hamiltonians:

$$\hat{X} := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \frac{1}{2} (X \otimes X + Y \otimes Y) = |10\rangle \langle 01| + |01\rangle \langle 10|, \quad (\text{B.58})$$

$$\hat{Y} := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \frac{1}{2} (Y \otimes X - X \otimes Y) = i |10\rangle \langle 01| - i |01\rangle \langle 10|. \quad (\text{B.59})$$

Note that $[X \otimes X, Y \otimes Y] = [X \otimes Y, Y \otimes X] = 0$, giving rise to a straightforward implementation of $e^{-i\hat{X}}$ and $e^{-i\hat{Y}}$ each using two parameterized two-qubit gates.

In particular, the subspace spanned by the one-hot code is an invariant subspace of \hat{X} and \hat{Y} . That is, for $1 \leq j < k \leq n$,

$$\frac{1}{2}(X_k \otimes X_j + Y_k \otimes Y_j) |h_j\rangle = |h_k\rangle, \quad \frac{1}{2}(X_k \otimes X_j + Y_k \otimes Y_j) |h_k\rangle = |h_j\rangle, \quad (\text{B.60})$$

$$\frac{1}{2}(X_k \otimes Y_j - Y_k \otimes X_j) |h_j\rangle = i |h_k\rangle, \quad \frac{1}{2}(X_k \otimes X_j - Y_k \otimes Y_j) |h_k\rangle = -i |h_j\rangle. \quad (\text{B.61})$$

Similar to the one-hot embedding with penalty, any matrix element is embedded with a 2-local operator. Indeed, for $1 \leq j < k \leq n$,

$$\langle h_j | \frac{(X_k \otimes X_j + Y_k \otimes Y_j)}{2} |h_k\rangle = 1, \quad \langle h_k | \frac{(X_k \otimes X_j + Y_k \otimes Y_j)}{2} |h_j\rangle = 1, \quad (\text{B.62})$$

$$\langle h_j | \frac{(X_k \otimes Y_j - Y_k \otimes X_j)}{2} |h_k\rangle = i, \quad \langle h_k | \frac{(X_k \otimes Y_j - Y_k \otimes X_j)}{2} |h_j\rangle = -i. \quad (\text{B.63})$$

Meanwhile, for $1 \leq j, k \leq n$, we observe that

$$\langle h_j | n_j |h_j\rangle = \begin{cases} 0 & (j \neq k) \\ 1 & (j = k). \end{cases} \quad (\text{B.64})$$

Proposition 17 (One-hot embedding without penalty). *Given an n -dimensional Hermitian matrix A , we define the following n -qubit quantum operator:*

$$H_A^{\text{ebd}} = \left(\sum_{j=1}^n \alpha_j \hat{n}_j \right) + \frac{1}{2} \left(\sum_{1 \leq j < k \leq n} \alpha_{j,k} (X_k \otimes X_j + Y_k \otimes Y_j) + \beta_{j,k} (X_k \otimes Y_j - Y_k \otimes X_j) \right), \quad (\text{B.65})$$

where α_j is the j -th diagonal entry of A , $\alpha_{j,k}$ and $\beta_{j,k}$ are the real and imaginary part of the (j, k) -th entry of A , respectively. Then, the 2-local quantum operator H_A^{ebd} is an embedding of A .

Proof. For any $j = 1, \dots, n$,

$$\langle h_j | H_A^{\text{ebd}} |h_j\rangle = \langle h_j | \left(\sum_{j=1}^n \alpha_j \hat{n}_j \right) |h_j\rangle = \alpha_j. \quad (\text{B.66})$$

For $1 \leq \ell < m \leq n$,

$$\langle h_\ell | H_A^{\text{ebd}} |h_m\rangle = \frac{1}{2} \langle h_\ell | \left(\sum_{1 \leq j < k \leq n} \alpha_{j,k} (X_k \otimes X_j + Y_k \otimes Y_j) + \beta_{j,k} (X_k \otimes Y_j - Y_k \otimes X_j) \right) |h_m\rangle \quad (\text{B.67})$$

$$= \frac{1}{2} \langle h_\ell | (\alpha_{\ell,m} (X_m \otimes X_\ell + Y_m \otimes Y_\ell) + \beta_{\ell,m} (X_m \otimes Y_\ell - Y_m \otimes X_\ell)) |h_m\rangle \quad (\text{B.68})$$

$$= \alpha_{\ell,m} + \beta_{\ell,m} i \quad (\text{B.69})$$

Since the encoding subspace for the one-hot code is an invariant subspace of H_A^{ebd} , no penalty Hamiltonian is needed. Thus, H_A^{ebd} is an embedding of A . \square

C Methodology of resource analysis

For each of the quantum simulation tasks considered, we formulate a target goal which corresponds to simulating the Hamiltonian for some time T , typically depending on the system size. We numerically compute an estimate of the resources (i.e. total gate count) required to perform each target task using Hamiltonian embedding. In all of the resource estimates considered, we do not consider any hardware specific error or additional overheads that may come from error correction.

As a baseline, we consider the naive approach of computing the Pauli decomposition of a Hamiltonian and performing simulation by Trotter-Suzuki product formulas. The Pauli decomposition of a Hamiltonian implicitly assumes the use of standard binary encoding to represent each computational basis state. The standard binary encoding is also the typical approach used extensively in quantum algorithms (for example, in the commonly used circuit for the quantum Fourier transform). When the system size N is not a power of 2, we start from the problem Hamiltonian to a larger one of dimension $2^{\lceil \lg N \rceil} \times 2^{\lceil \lg N \rceil}$ and remove any off-diagonal terms corresponding to edges between the target subgraph and the unused component. When the problem Hamiltonian has nontrivial diagonal terms, we additionally modify the diagonal terms to ensure a correct encoding of the desired problem Hamiltonian.

For each task experimentally demonstrated on the IonQ processor, we compare the estimated gate count of the standard binary encoding with other Hamiltonian embedding schemes. For a fair comparison, we fix the same Hamiltonian simulation method (i.e. the choice of product formula) and the desired precision across the different embedding methods. For the standard binary encoding, we use Qiskit [20] to compute the Pauli decomposition of the (possibly modified) problem Hamiltonian and compile the circuit to one- and two-qubit gates. In addition to optimizing the circuit using the Qiskit transpiler, we further optimize the circuit using TKET [105]. The basis gate set is chosen to include parameterized single-qubit Pauli rotations and XX -rotation gates: $\{R_x(\theta), R_y(\theta), R_z(\theta), R_{xx}(\theta)\}$. The additive Trotter error is estimated by Monte Carlo sampling. In all resource estimates performed, we use binary search to numerically compute the Trotter number to achieve error within $\epsilon = 5 \times 10^{-2}$. For reference, we also include theoretical estimates of the gate count, computed using the worst-case Trotter error bound for the unary and one-hot embeddings [100, 34]. We fit the data to power laws to obtain estimates of the asymptotic gate complexity.

For the experiments on QuEra, we do not perform a resource comparison between the standard binary encoding and the use of Hamiltonian embedding for simulating real-space dynamics. The programmability of analog quantum simulators naturally forbids the use of dense encodings (such as the standard binary encoding and the Gray code) to simulate the desired problem Hamiltonian directly. For the standard binary code, adjacent vertices in the graph of the problem Hamiltonian may have Hamming distance larger than 1 (for instance, bitstrings 0111 and 1000 representing 7 and 8, respectively). On the other hand, the Gray code suffers from the issue of having unwanted interactions, even if the Hamming distance is small (for instance, bitstrings 0111 and 1111 representing 7 and 15, respectively).

D Quantum walk on graphs

Given a graph $G = (V, E)$, the continuous-time quantum walk on the graph G is described by the Schrödinger equation,

$$i \frac{d}{dt} |\psi\rangle = L |\psi\rangle, \quad (\text{D.1})$$

where L is the graph Laplacian defined by:

$$(L)_{j,k} = \begin{cases} 0 & (j, k) \notin E \\ 1 & (j, k) \in E \\ -\deg(j) & j = k \end{cases} \quad (\text{D.2})$$

In this section, we construct Hamiltonian embeddings of the graph Laplacian L for several important families of graphs, including finite-dimensional lattices and trees.

D.1 Lattice graphs

In this section, we discuss the embedding of d -dimensional Lattice graphs with two types of boundary conditions: the regular lattice graphs are grids with open boundaries, and the periodic lattice graphs have periodic boundary conditions (i.e., torus-like shape). For both types of graphs, the size of the graph Laplacian L scales exponentially in dimension d . By leveraging a tensor-product decomposition of L , we construct 2-local Hamiltonian embeddings of the graph Laplacian with $O(d)$ qubits.

D.1.1 Regular lattice graphs

Definition 7 (Regular lattice graphs). Fix an integer $N \geq 1$, the d -dimensional regular lattice graph $\mathcal{G}_{\text{reg}} = (\mathcal{V}_{\text{reg}}, \mathcal{E}_{\text{reg}})$ has N^d vertices:

$$\mathcal{V}_{\text{reg}} = \{\mathbf{v} = (v_1, v_2, \dots, v_d) : v_1, \dots, v_d \in \{1, \dots, N\}\}, \quad (\text{D.3})$$

and there is an edge between \mathbf{u} and \mathbf{v} if and only if their coordinates differ by 1 at a single site. In other words, $(\mathbf{u}, \mathbf{v}) \in \mathcal{E}_{\text{reg}}$ if there exists an index $k \in \{1, \dots, d\}$ such that $|v_k - u_k| = 1$ and $v_j = u_j$ for all $j \neq k$.

Regular lattice graphs are sometimes referred to as grid graphs. The graph Laplacian of the d -dimensional regular lattice graph is given by

$$L_{\text{reg}} = \sum_{k=1}^d I \otimes \dots \otimes \underbrace{L_{\text{chain}}}_{\text{the } k\text{-th operator}} \otimes \dots \otimes I, \quad (\text{D.4})$$

where L_{chain} is the graph Laplacian of the 1D chain graph (with N nodes):

$$L_{\text{chain}} = \begin{bmatrix} -1 & 1 & & \\ 1 & -2 & 1 & \\ \dots & \dots & \dots & \dots \\ & 1 & -2 & 1 \\ & & 1 & -1 \end{bmatrix} \quad (\text{D.5})$$

From (D.4), it is clear that the graph Laplacian L_{reg} is the composition of d independent copies of L_{chain} . Since the Laplacian of the 1D chain graph is a tridiagonal matrix, we can

use an embedding scheme in [Appendix B.1](#) or the one-hot embedding⁸ (see [Appendix B.3](#)) to construct an embedding of L_{chain} .

1. Unary embedding of L_{chain} :

$$H^{\text{pen}} = - \sum_{j=1}^{N-2} Z_{j+1} Z_j + Z_1 - Z_{N-1}, \quad Q = (-I - \hat{n}_1 + \hat{n}_{N-1}) + \sum_{j=1}^{N-1} X_j.$$

2. Antiferromagnetic embedding of L_{chain} :

$$H^{\text{pen}} = \sum_{j=1}^{N-2} Z_{j+1} Z_j + Z_1 + (-1)^{N-1} Z_{N-1}, \quad Q = -2(N-1)I - \hat{n}_1 + (-1)^N \hat{n}_{N-1} + \sum_{j=1}^{N-1} X_j.$$

3. One-hot embedding of L_{chain} (with penalty):

$$H^{\text{pen}} = \left(\sum_{j=1}^N \hat{n}_j - 1 \right)^2, \quad Q = \left(-\hat{n}_1 - \hat{n}_N - 2 \sum_{j=2}^{N-1} \hat{n}_j \right) + \sum_{j=1}^{N-1} X_{j+1} X_j.$$

4. Penalty-free one-hot embedding of L_{chain} :

$$H^{\text{ebd}} = \left(-\hat{n}_1 - \hat{n}_N - 2 \sum_{j=2}^{N-1} \hat{n}_j \right) + \sum_{j=1}^{N-1} (X_{j+1} X_j + Y_{j+1} Y_j).$$

Next, we use [Lemma 6](#) to construct an embedding of L_{reg} using d independent embeddings of L_{chain} . In [Table 12](#), we summarize the resources required in different Hamiltonian embeddings of L_{reg} .

Embedding scheme	# of qubits	# of 2-local operators	Max. weight
Unary	$d(N-1)$	$d(N-2)$	2
Antiferromagnetic	$d(N-1)$	$d(N-2)$	2
One-hot w/ penalty	dN	$d(N-1)(N+2)/2$	2
Penalty-free one-hot	dN	$2d(N-1)$	2

Table 12: Resource count for various Hamiltonian embeddings of L_{reg} .

D.1.2 Periodic lattice graphs

Definition 8 (Periodic lattice graphs). Fix an integer $N \geq 1$, the d -dimensional periodic lattice graph $\mathcal{G}_{\text{prd}} = (\mathcal{V}_{\text{prd}}, \mathcal{E}_{\text{prd}})$ has N^d vertices:

$$\mathcal{V}_{\text{prd}} = \{ \mathbf{v} = (v_1, v_2, \dots, v_d) : v_1, \dots, v_d \in \mathbb{Z}_N \}, \quad (\text{D.6})$$

and there is an edge between \mathbf{u} and \mathbf{v} if and only if there exists an index $k \in \{1, \dots, d\}$ such that $v_k = u_k \pm 1$ and $v_j = u_j$ for all $j \neq k$.

⁸It is worth noting that we must use the penalty Hamiltonian defined in [\(B.43\)](#) because [Lemma 6](#) requires the embedding subspace to be the ground-energy subspace of H^{pen} .

The graph Laplacian of the d -dimensional periodic lattice graph is given by

$$L_{\text{prd}} = \sum_{k=1}^d I \otimes \cdots \otimes \underbrace{L_{\text{cycle}}}_{\text{the } k\text{-th operator}} \otimes \cdots \otimes I, \quad (\text{D.7})$$

where L_{cycle} is the graph Laplacian of the 1D cycle graph (with N nodes):

$$L_{\text{cycle}} = \begin{bmatrix} -2 & 1 & & 1 \\ 1 & -2 & 1 & \\ \cdots & \cdots & \cdots & \cdots \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{bmatrix} \quad (\text{D.8})$$

Similarly, the graph Laplacian L_{prd} is the composition of d independent copies of L_{cycle} . We observe that L_{cycle} is a tridiagonal circulant matrix, so we can use an embedding scheme in [Appendix B.2](#) or the one-hot embedding to construct an embedding of L_{cycle} . Note that we omit the -2 global phase in the following embeddings.

1. Circulant unary embedding of L_{cycle} (N must be an even number):

$$H^{\text{pen}} = - \sum_{j=1}^{N/2-1} Z_{j+1}Z_j + Z_{N/2}Z_1, \quad Q = \sum_{j=1}^{N/2} X_j.$$

2. Circulant antiferromagnetic embedding of L_{cycle} (N must be an even number):

$$H^{\text{pen}} = \sum_{j=1}^{N/2-1} Z_{j+1}Z_j - (-1)^{N/2} Z_{N/2}Z_1, \quad Q = \sum_{j=1}^{N/2} X_j.$$

3. One-hot embedding of L_{cycle} (with penalty, N can be even or odd):

$$H^{\text{pen}} = \left(\sum_{j=1}^N \hat{n}_j - 1 \right)^2, \quad Q = \sum_{j=1}^{N-1} X_{j+1}X_j + X_N X_1.$$

4. Penalty-free one-hot embedding of L_{cycle} :

$$H^{\text{ebd}} = \sum_{j=1}^{N-1} (X_{j+1}X_j + Y_{j+1}Y_j) + X_N X_1 + Y_N Y_1.$$

An embedding of L_{prd} can be obtained by [Lemma 6](#). In [Table 13](#), we summarize the resources required in different Hamiltonian embeddings of L_{prd} .

D.1.3 Experiments on IonQ

We conduct experiments for quantum walk on both the 1D chain and 1D cycle graphs on the IonQ Aria-1 processor.

For both the 1D chain and 1D cycle, we demonstrate quantum walk on $N = 15$ vertices with the penalty-free one-hot embedding. The initial state is taken to be the middle vertex in the case of the regular lattice, prepared using a single one-qubit gate. Due to symmetry, the choice of the vertex does not matter for quantum walk on a periodic lattice. The

Embedding scheme	# of qubits	# of 2-local operators	Max. weight
Circ-Unary	$dN/2$	$dN/2$	2
Circ-Antiferromagnetic	$dN/2$	$dN/2$	2
One-hot w/ penalty	dN	$dN(N + 1)/2$	2
Penalty-free one-hot	dN	$2dN$	2

Table 13: Resource count for various Hamiltonian embeddings of L_{prd} .

maximum evolution time is taken to be $T = 4.0$ with $r = 5$ Trotter steps using the randomized first-order Trotter-Suzuki formula. The two-qubit gate counts correspond to the number of 2-local operators, as shown in Table 12. For each circuit, the two-qubit gate counts are 140 and 150 for the 1D chain and 1D cycle, respectively.

For the experimental demonstrations of quantum walk on lattices, we adopt the definition of continuous time quantum walk using the adjacency matrix of the graph, rather than the graph Laplacian. The choice of using the adjacency matrix is merely for simplicity, as it avoids introducing additional Pauli- Z rotations. Since the lattice graph without periodic boundary conditions is not a regular graph, this gives rise to different dynamics. For the periodic lattice, the adjacency matrix differs from the Laplacian by a constant, so the dynamics are equivalent.

In panels A, B, D, and E of Figure 5, we show experimental results simulating quantum walk on the 1D chain and cycle, each with 15 vertices. Due to the initial state, the dynamics are qualitatively similar between the chain and lattice. Experimentally, we observe the expected propagation distance increases linearly with time, closely matching the numerically predicted result.

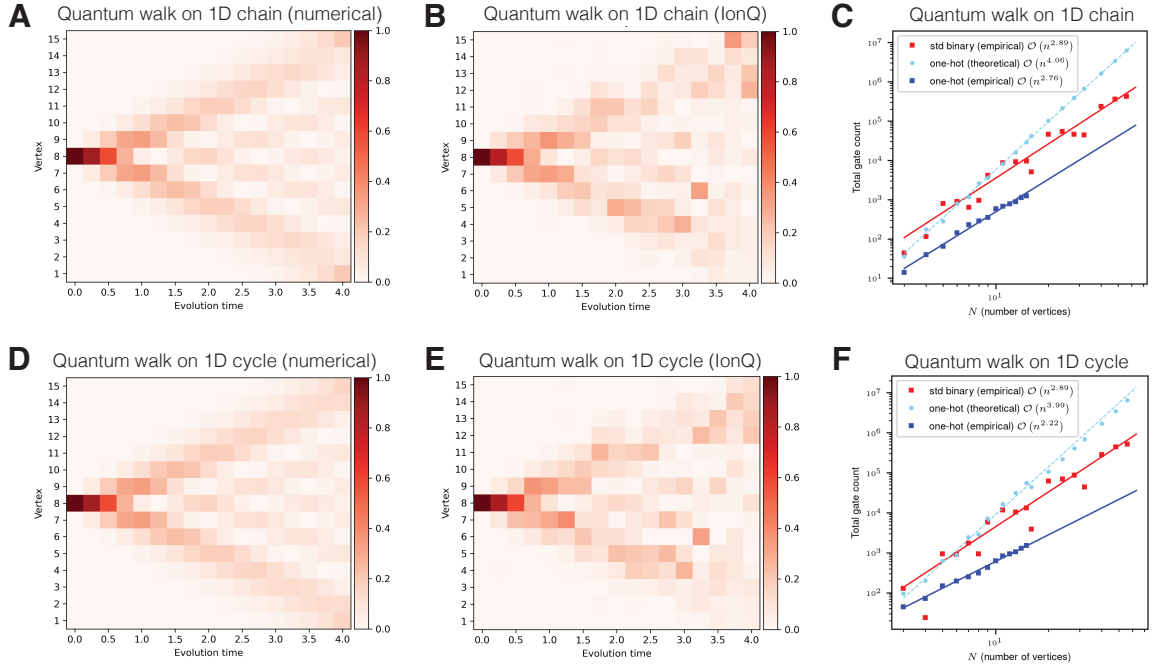


Figure 5: Quantum walk on 1-dimensional regular lattice graph (chain) and periodic lattice graph (cycle) of order $N = 15$.

D.1.4 Resource analysis

We conduct a resource comparison between the standard binary code and Hamiltonian embedding to perform quantum walk on one-dimensional lattice graphs. We assume the initial state is the first vertex v_1 on the chain or cycle and define the task as follows.

We define the expected propagation distance observable as

$$O_{\text{prop}} = \sum_{j=1}^N d(v_1, v_j) |v_j\rangle \langle v_j|, \quad (\text{D.9})$$

where $d(v_1, v_j)$ is the distance between v_1 and v_j in the graph, and $|v_j\rangle$ denotes the encoded state for vertex v_j . The task is then to evolve the system for time T_N such that the propagation distance is at least $N/4$, i.e.

$$T_N = \min_{t>0} \{t \mid \langle \psi(t) | O_{\text{prop}} | \psi(t) \rangle > N/4\}, \quad (\text{D.10})$$

where $|\psi(t)\rangle$ is the wave function at time t as defined in (D.1). In panels C and F of Figure 5, we show estimates of the total gate count required to simulate quantum walk on regular and periodic lattice graphs of order N for evolution time T_N using standard binary and the penalty-free one-hot embedding. As in the experiments, we use the randomized first-order Trotter formula to simulate the Hamiltonian. In both cases, the empirical data suggests that the penalty-free one-hot embedding has better constant factors as well as slightly better asymptotic scaling.

D.2 Binary tree and glued trees graph

A binary tree is a graph in which each node has at most two children. A glued trees graph consists of two copies of binary trees in which leaf nodes are “glued” together randomly to form a cycle. Binary tree and the glued trees graph are important structures in computer science. In particular, the glued trees graph is of significant theoretical interest because the problem of traversing the graph has been shown to exhibit an exponential separation between classical and quantum algorithms [68]. In this section, we discuss the embedding of perfect binary trees and glued trees graphs.

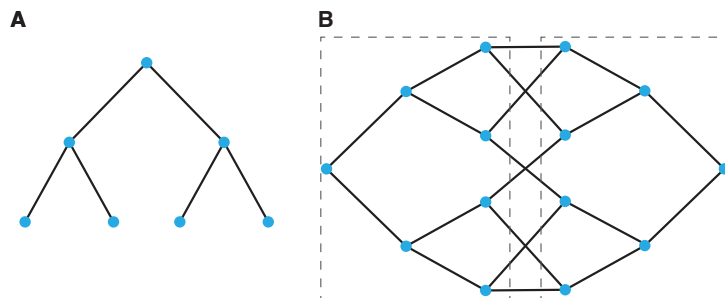


Figure 6: **A.** A height-2 perfect binary tree. **B.** A glued trees graph consisting of two height-2 binary trees.

D.2.1 Binary tree

A perfect height- h binary tree has $2^{h+1} - 1$ nodes. In [Figure 6A](#), we illustrate a binary tree of height 2. The adjacency matrix of this tree is of dimension 7,

$$A_{\text{tree}} = \begin{bmatrix} & 1 & 1 & & & & \\ 1 & & & 1 & 1 & & \\ 1 & & & & & 1 & 1 \\ & 1 & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & 1 & & & & \end{bmatrix} \quad (\text{D.11})$$

The adjacency matrix A_{tree} does not have a band or circulant structure. We use the one-hot embedding (see [Appendix B.3](#)) to construct an embedding of A_{tree} . To simplify the notation, we define the set of edges of the binary tree $E_{\text{tree}} = \{(1, 2), (1, 3), (2, 4), (2, 5), (3, 6), (3, 7)\}$.

1. One-hot embedding (with penalty) of A_{tree} : $H^{\text{pen}} = \sum_{j=1}^7 Z_j$, $Q = \sum_{(j,k) \in E} X_j X_k$.
2. Penalty-free one-hot embedding of A_{tree} : $H^{\text{ebd}} = \frac{1}{2} \left(\sum_{(j,k) \in E} X_j X_k + Y_j Y_k \right)$.

In [Table 14](#), we summarize the resources required in different Hamiltonian embeddings of A_{tree} .

Embedding scheme	# of qubits	# of 2-local operators	Max. weight
One-hot w/ penalty	$2^{h+1} - 1$	$2^{h+1} - 2$	2
Penalty-free one-hot	$2^{h+1} - 1$	$2^{h+2} - 4$	2

Table 14: Resource count for one-hot embeddings of height- h binary trees.

D.2.2 Glued trees graph

In [Figure 6B](#), we show a 6-level glued trees graph (consisting of 2 height-2 binary trees). The adjacency matrix of the glued trees graph is $A_{\text{glued}} = \begin{bmatrix} A_{\text{tree}} & W \\ W^\dagger & A_{\text{tree}} \end{bmatrix}$, where W is a 7-by-7 matrix representing the connectivity pattern between the two trees,

$$W = \begin{bmatrix} & & & & & & \\ 1 & 1 & & & & & \\ 1 & & 1 & & & & \\ & 1 & & 1 & & & \\ & & & 1 & 1 & & \end{bmatrix}$$

Since the full adjacency matrix A_{glued} is a 14-by-14 sparse matrix, we can construct Hamiltonian embeddings of A_{glued} using one-hot embedding. In the standard one-hot embedding (with penalty), we need to use 14 qubits and 20 two-qubit operators (all XX rotations). In the penalty-free one-hot embedding, we need 14 qubits and 40 two-qubit operators (20 XX plus 20 YY rotations).

D.2.3 Experiments on IonQ

We experimentally demonstrate quantum walk on the binary tree and glued trees graph using the IonQ Aria-1 processor. In both cases, we use the penalty-free one-hot embedding and we choose to embed the adjacency matrix rather than the Laplacian. This choice of Hamiltonian avoids the need for additional Pauli- Z rotations required to embed the diagonal terms of the Laplacian. In both cases, we demonstrate traversal through the graphs by choosing the initial state to be the root node (in the case of the binary tree) or the entrance node (in the case of the glued trees graph).

The binary tree is chosen to be a perfect binary tree with height 3, i.e. $N = 15$ nodes. The maximum evolution time is $T = 3.0$ with $r = 6$ Trotter steps. Each circuit consists of 168 two-qubit gates and a single one-qubit gate for preparing the root node state. In [Figure 7A](#), we plot the propagation of the wavefunction across the levels of the binary tree, starting from the root (denoted as level 1).

For quantum walk on the glued trees graph, we use two binary trees each of height 2, resulting in a 14-node glued trees graph as shown in [Figure 6](#). The maximum evolution time is taken to be $T = 2.0$ with $r = 4$ Trotter steps. Each circuit consists of 120 two-qubit gates and a single one-qubit gate for preparing the entrance node state. In both cases, we use the randomized first-order Trotter-Suzuki formula to simulate the Hamiltonian.

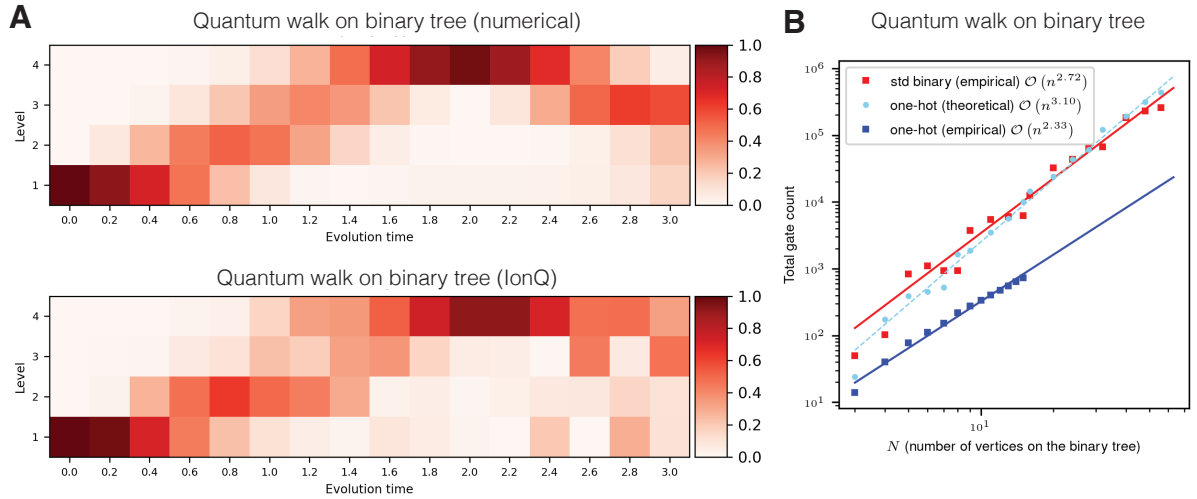


Figure 7: Quantum walk on a binary tree of height 3. **A.** Heatmap of the quantum walk evolution (clustered on each layer of the binary tree). **B.** Estimated total gate count for quantum walk on binary trees with N nodes. For standard binary, we estimate the gate count for all integers N ranging from 3 to 63, but only a selection of points are plotted for visualization.

D.2.4 Resource analysis

In this section, we perform a resource analysis for the task of traversing the binary tree and glued trees graph.

For the binary tree, we are interested in the propagation O_{prop} defined in exactly the same manner as in [Appendix D.1.4](#). Here, we set the target distance to be equal to half of the height of the binary tree, so that

$$T_N = \min_{t>0} \left\{ t \mid \langle \psi(t) | O_{\text{prop}} | \psi(t) \rangle > \frac{\lfloor \lg N \rfloor}{2} \right\}. \quad (\text{D.12})$$

For the glued tree, we are primarily interested in the traversal from the entrance node to the exit node, which we define as follows. Let G_h be a glued trees graph with $2(h+1)$ layers (h being the height of each of the binary trees), and let A_h be the adjacency matrix. As defined in [68], let $|\text{col } j\rangle$ be the uniform superposition over vertices in the j -th layer, i.e.

$$|\text{col } j\rangle = \frac{1}{\sqrt{N_j}} \sum_{a \in \text{layer } j} |a\rangle, \quad (\text{D.13})$$

where

$$N_j = \begin{cases} 2^j & 0 \leq j \leq h, \\ 2^{2h+1-j} & h+1 \leq j \leq 2h+1. \end{cases} \quad (\text{D.14})$$

for $j = 0, 1, \dots, 2h+1$. For a constant $s \in (0, 1)$, let $T_{h,s}$ be the evolution time needed for the probability on the exit node to be at least s , i.e.,

$$T_{h,s} = \min_{t>0} \left\{ t : \left| \langle \text{col } 0 | e^{-itA_h} | \text{col } 2h+1 \rangle \right|^2 \geq s \right\}, \quad (\text{D.15})$$

where the initial state is the entrance node $|\text{col } 0\rangle$.

In Figure 7B, we show estimated total gate counts for quantum walk on the binary tree and glued trees graphs. The target evolution time is set to $T_{h,s}$ with $s = 0.4$. As in the experimental demonstrations, we use the randomized first-order product formula to simulate the Hamiltonian. Empirical estimates suggest that one-hot embedding requires significantly fewer gates than the standard binary code, with a slightly better asymptotic scaling.

E Spatial search on regular lattices

The spatial search problem is to find a marked item in a structured database. Precisely, given an unweighted graph $G = (\mathcal{V}, \mathcal{E})$ without self loops, and an oracle function $f: \mathcal{V} \rightarrow \{0, 1\}$ that returns 1 if the input vertex x is *marked* and 0 otherwise. For simplicity, we assume there is only one marked vertex \mathbf{v} , and the oracle function f is given by

$$f(x) = \begin{cases} 0 & (x \neq \mathbf{v}) \\ 1 & (x = \mathbf{v}). \end{cases} \quad (\text{E.1})$$

The spatial search problem requires finding the marked vertex \mathbf{v} by querying the oracle f .

While no classical algorithms can do better than exhaustive search, quantum algorithms can provide speedups for this problem [106, 80, 107]. Childs and Goldstone [80] proposed a quantum search algorithm by continuous-time quantum walk, which demonstrates quantum speedups for spatial search on various graphs. This quantum algorithm is formulated as a continuous-time quantum walk:

$$H_{\text{search}} = -\gamma L - H_{\mathbf{v}}, \quad (\text{E.2})$$

where L is the graph Laplacian of G (for details, see Appendix D), and the *oracle Hamiltonian* is defined as

$$H_{\mathbf{v}} = |\mathbf{v}\rangle \langle \mathbf{v}|. \quad (\text{E.3})$$

The parameter $\gamma > 0$ is chosen to minimize the spectral gap of H .

To simulate the Hamiltonian H_{search} on quantum computers, we need to build a Hamiltonian embedding for it. In this section, we consider the spatial search problem on a d -dimensional *regular lattice graph*, whose graph Laplacian L_{reg} can be embedded using a Hamiltonian embedding scheme proposed in [Appendix D.1.1](#). In what follows, we focus on the embedding of the oracle Hamiltonian $H_{\mathbf{v}}$.

E.1 Embedding of the oracle Hamiltonian

We exemplify the embedding of the oracle Hamiltonian $H_{\mathbf{v}}$ using the unary embedding scheme. The same idea can be readily generalized to antiferromagnetic embedding and one-hot embedding.

First, we discuss how to embed the oracle Hamiltonian for a 1D chain graph with N vertices $\{v_1, \dots, v_N\}$:

1. If the marked vertex is v_1 , the unary codeword for v_1 is the $(N-1)$ -bit all-zero string $|0\dots 0\rangle$. This is the only unary codeword with the first bit 0, so we can construct an embedding of H_{v_1} by marking the first qubit: $H_{v_1}^{\text{ebd}} = I - \hat{n}_1$.
2. Similarly, if the marked vertex is v_N , the unary codeword for v_N is the $(N-1)$ -bit all-one string $|1\dots 1\rangle$. This is the only unary codeword with the last bit 1, so we can construct an embedding of H_{v_N} by marking the last qubit: $H_{v_N}^{\text{ebd}} = \hat{n}_{N-1}$.
3. If the marked vertex is among $\{v_2, \dots, v_{N-1}\}$, i.e., in the interior of the chain, we can distinguish this vertex from others by its $(N-1)$ -th and N -th bits. In particular, v_j is the only unary codeword whose $(N-1)$ -th bit is 1 and N -th bit is 0. Therefore, we can construct an embedding of H_{v_j} by marking its $(N-1)$ -th and N -th bits: $H_{v_j}^{\text{ebd}} = (I - \hat{n}_j) \otimes \hat{n}_{j-1}$.

In summary, the oracle Hamiltonian for the spatial search on the 1D chain is at most 2-local. Based on the embedding of the oracle Hamiltonian for 1D chain, we can construct embeddings of the oracle Hamiltonian for any d -dimensional regular lattice:

$$H_{\mathbf{x}}^{\text{ebd}} = H_{x_1}^{\text{ebd}} \otimes H_{x_2}^{\text{ebd}} \otimes \dots \otimes H_{x_d}^{\text{ebd}}, \quad (\text{E.4})$$

where the vertex $\mathbf{x} = (x_1, \dots, x_d)$, each $x_k \in \{v_1, \dots, v_N\}$, and we define

$$H_{x_k}^{\text{ebd}} = \begin{cases} I - \hat{n}_1 & (x_k = v_1) \\ \hat{n}_{N-1} & (x_k = v_N) \\ (I - \hat{n}_j) \otimes \hat{n}_{j-1} & (x_k = v_j) \end{cases} \quad (\text{E.5})$$

It turns out that the embedding of the oracle Hamiltonian $H_{\mathbf{x}}$ is at least d -local and at most $2d$ -local.

Remark 3. Similar ideas can be used to mark vertices on the edge and interior of a 2D lattice graph. However, we need a 3- or 4-tensor operator to realize the oracle Hamiltonian because we need 3 qubits to unambiguously distinguish an edge vertex, and 4 qubits for an interior vertex. For example, on a 4-by-4 lattice, the operator that marks the vertex $\mathbf{v} = (2, 4)$ is $H' = \hat{n}_1 \hat{n}_2 \hat{n}_6$, and the operator that marks the vertex $\mathbf{v} = (3, 3)$ is $H' = (I - \hat{n}_2)(I - \hat{n}_3)(I - \hat{n}_5)(I - \hat{n}_6)$.

Node	1	2	3	4	5	6
Code word	$g_1 = 010$	$g_2 = 110$	$g_3 = 100$	$g_4 = 101$	$g_5 = 001$	$g_6 = 011$

Table 15: Code words for the 4-node cycle graph.

Generalization to periodic lattices. Similarly, a marked vertex on the periodic lattice can be perfectly distinguished by a few bits. Consider the codewords for the cycle graph $\text{CW-cycle}(n)$ with $n = 3$, we have: There are no “endpoints” on the cycle graph, and each vertex codeword is distinguished from the others by exactly 2 bits. For example, g_1 is the only codeword with the first and last bit being 0, and g_2 is the only codeword with the first and second bit being 1, etc. This implies that we need a 4-tensor to encode the oracle Hamiltonian $H_{\mathbf{v}}$ for any vertex \mathbf{v} on the 2D periodic lattice. Again, such operators can not be engineered on neutral-atom quantum computers.

E.2 Experiments on IonQ

E.2.1 Experiment setup

We perform experimental demonstrations of spatial search by continuous time quantum walk using Hamiltonian embedding. We present two spatial search experiments: the first being on a 4×4 regular lattice with the unary embedding, and the second being on a 5×5 regular lattice with the penalty-free one-hot embedding.

The algorithm for spatial search by continuous time quantum walk requires the initial state to be in a uniform superposition. For the standard binary encoding, this is trivially done using a layer of Hadamard gates. However, initial state preparation becomes nontrivial when considering alternate encodings such as unary and one-hot.

We design state preparation circuits for both the unary and one-hot codes. In [Algorithm 2](#) and [Algorithm 3](#), we show the construction of state preparation circuits for the unary and one-hot encodings, assuming real-valued amplitudes. The idea behind these circuits is to start from the first encoded state and sequentially rotate probability amplitude using two-qubit gates on consecutive qubits. In the case of complex-valued amplitudes, one may introduce local phases by an additional sequence of diagonal gates, such as single Z rotations or ZZ rotations.

Algorithm 2: 1D state preparation with unary encoding

Input: Real amplitudes a_1, \dots, a_{n+1} , input state $|00 \dots 0\rangle$

Output: Quantum state $\sum_{k=1}^{n+1} a_k |w_k\rangle$

1 Apply $R_y(2 \arccos(a_1))$ to qubit 1.

2 **for** $k = 1, \dots, n$ **do**

3 Let $\alpha_{k+1} = a_{k+1} \left(\sum_{j=k+1}^{n+1} |a_j|^2 \right)^{-1/2}$.

4 Apply $R_y(2 \arccos(\alpha_{k+1}))$ to qubit $k + 1$ controlled on qubit k .

We note that a version of the state preparation circuit for the one-hot code has appeared in [\[104\]](#), referred to as a “diagonal unary loader” in the context of quantum neural networks. For the one-hot code specifically, the circuit can additionally be parallelized to achieve $O(\log n)$ circuit depth. In practice, we implement the parallel version involving the partial swap gate by using two arbitrary-angle MS gates, resulting in fewer gates.

For both the unary and one-hot embeddings, we mark the corner vertex $\mathbf{x} = (N, 1)$

Algorithm 3: 1D state preparation with one-hot encoding, sequential version

Input: Real amplitudes a_1, \dots, a_n , input state $|10\dots 0\rangle$
Output: Quantum state $\sum_{k=1}^n a_k |w_k\rangle$
 1 Apply $R_y(2 \arccos(a_1))$ to qubit 2.
 2 Apply CNOT to qubit 1 controlled on qubit 2.
 3 **for** $k = 1, \dots, n - 1$ **do**
 4 Let $\alpha_{k+1} = a_{k+1} \left(\sum_{j=k+1}^n |a_j|^2 \right)^{-1/2}$.
 5 Apply $R_y(2 \arccos(\alpha_{k+1}))$ to qubit $k + 1$ controlled on qubit k .
 6 Apply CNOT to qubit k controlled on qubit $k + 1$.

when constructing the embedding of the oracle Hamiltonian. This results in a single $\hat{n}_{N-1} \otimes (I - \hat{n}_1)$ term for the unary embedding and a $\hat{n}_N \otimes \hat{n}_1$ term for the one-hot embedding.

For spatial search with the unary embedding, we set $N = 4$, for a total of 6 qubits. For the unary embedding, the embedding Hamiltonian is given by

$$H_{\text{unary}}^{\text{ebd}} = -\gamma H_L^{\text{ebd}} - H_{\mathbf{x}}^{\text{ebd}} + gH^{\text{pen}}, \quad (\text{E.6})$$

where H_L^{ebd} and $H_{\mathbf{x}}^{\text{ebd}}$ are embeddings of the graph Laplacian and oracle Hamiltonian, respectively, and H^{pen} is the penalty Hamiltonian for the unary embedding. For the real-machine experiment, we set the penalty parameter to $g = 2$. The embedding Hamiltonian consists of two-qubit ZZ terms, as well as additional single-qubit X and Z terms.

For the one-hot embedding, the embedding Hamiltonian is given by

$$H_{\text{one-hot}}^{\text{ebd}} = -\gamma H_L^{\text{ebd}} - H_{\mathbf{x}}^{\text{ebd}}, \quad (\text{E.7})$$

where H_L^{ebd} and $H_{\mathbf{x}}^{\text{ebd}}$ are embeddings of the graph Laplacian and oracle Hamiltonian. No penalty term is needed due to the use of the penalty-free embedding. Here, the embedding Hamiltonian consists of two-qubit XX , YY , and ZZ terms as well as additional single-qubit Z terms to encode the diagonal entries of the graph Laplacian.

In both cases, we simulate the Hamiltonian using the second-order Trotter-Suzuki formula. For spatial search on a 4×4 lattice with the unary encoding, we perform simulation for unitless time up to $T \approx 4.894$ with $r = 12$ Trotter steps. The largest resulting circuit consists of 6 qubits, 116 one-qubit gates, and 112 two-qubit gates. For the 5×5 lattice with the penalty-free one-hot embedding, we perform simulation for unitless time $T \approx 6.569$ with $r = 5$ Trotter steps. Each circuit consists of 10 qubits, 22 one-qubit gates and 181 two-qubit gates. The evolution times are chosen to achieve a desired threshold success probability and is described in more detail in [Appendix E.2.2](#) as part of the resource analysis.

E.2.2 Resource analysis

We conduct a resource analysis comparing the standard binary encoding with the unary and penalty-free one-hot embeddings. In all cases, we consider spatial search on a d -dimensional lattice with N vertices along each axis. Starting from a uniform initial state, we define the success probability to be $|\langle \psi(t) | \mathbf{v} \rangle|^2$. The spatial search task is to evolve the initial state according to the Hamiltonian H_{search} for time

$$T_p = \min_{t>0} \{t : |\langle \psi(t) | \mathbf{v} \rangle|^2 \geq p\}, \quad (\text{E.8})$$

such that the success probability is at least p . Note that for dimension $d = 2$, the success probability is bounded by $O((\log(N)/N)^2)$ [80], so we choose $p = 4(\log(N)/N)^2$. For reference, $T_p \approx 4.89$ for $N = 4$ and $T_p \approx 6.57$ for $N = 5$. The parameter γ is numerically computed by minimizing the spectral gap using the Nelder-Mead method.

For the standard binary encoding, we start with the spatial search Hamiltonian for $2^{\lceil \lg N \rceil}$ vertices, where N is the desired number of vertices along a single axis and d is the dimension of the lattice. To properly encode the Laplacian of the d -dimensional lattice with N^d vertices, we isolate the subgraph with N^d vertices, removing all off-diagonal terms that contribute to interactions between the subgraph and the remaining component of the graph. The diagonal terms of the target submatrix are modified to match the spatial search Hamiltonian. This procedure helps preserve the tensor product structure of the target Hamiltonian.

For the unary embedding, we choose the penalty parameter $g = \gamma ndT_p$, where γ is the parameter for the oracle Hamiltonian. This choice of penalty parameter yields a unitary fidelity of over 0.99 for all system sizes tested. Here, we define the unitary fidelity restricted to the encoding subspace by

$$f = \frac{1}{Nd} \left\| \text{tr} \left[U_{\text{search}}^\dagger (P_S U_{\text{ebd}} P_S) \right] \right\|, \quad (\text{E.9})$$

where U_{search} and U_{ebd} are the evolution operators for H_{search} and H^{ebd} , respectively.

For both the unary and one-hot embedding, we apply the second-order Trotter-Suzuki product formula and use Monte Carlo sampling to evaluate the additive Trotter error.

In Figure 3B we show the estimated gate counts for performing spatial search on $N \times N$ lattices. The marked vertex is $\mathbf{x} = (N, 1)$, so the embedded oracle Hamiltonian is $\hat{n}_{N-1} \otimes (I - \hat{n}_1)$. We test system sizes up to $N = 11$ for standard binary and up to $N = 6$ for the unary and penalty-free one-hot embeddings. While the unary embedding requires fewer gates empirically than standard binary for the system sizes tested, the asymptotic scaling is slightly worse. On the other hand, the penalty-free one-hot embedding shows a significant advantage both in the empirically estimated gate counts and the asymptotic scaling.

We additionally show resource estimates on 3D lattices in Figure 8. In this case, the marked vertex is set to $\mathbf{x} = (N, N, N)$, resulting in the embedded oracle Hamiltonian to be $\hat{n}_N \otimes \hat{n}_N \otimes \hat{n}_N$. We observe advantages for the unary and penalty-free one-hot embeddings in both the constant factor and asymptotic scaling, according to the empirical estimates.

F Simulating quantum dynamics in the real space

In this section, we consider the simulation of quantum dynamics in the real space. The quantum dynamics is described by the Schrödinger equation over \mathbb{R}^d :

$$i \frac{\partial}{\partial t} \Psi = \left[-\frac{1}{2} \nabla^2 + f(x) \right] \Psi(t, x), \quad (\text{F.1})$$

where $\Psi(t, x): [0, T] \times \mathbb{R}^d \rightarrow \mathbb{C}$ is the quantum wave function. The Schrödinger equation is subject to an initial condition $\Psi(0, x) = \Psi_0(x)$.

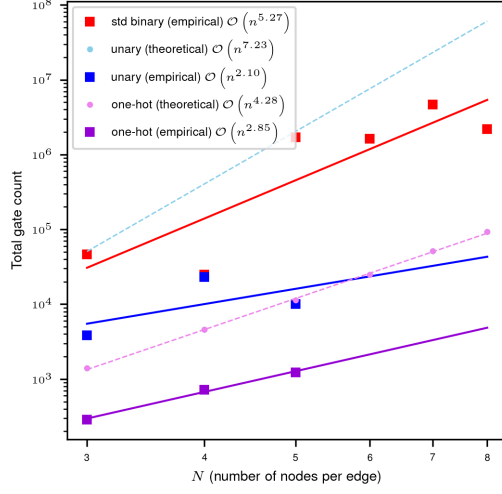


Figure 8: Estimated total gate count for spatial search on 3D lattice.

F.1 Experiments on IonQ

F.1.1 Problem formulation

Lemma 18. Consider the 1D Schrödinger equation with quadratic potential field ($a > 0$):

$$i \frac{\partial}{\partial t} \Psi(t, x) = \left[-\frac{1}{2} \frac{\partial^2}{\partial x^2} + \left(\frac{1}{2} a x^2 + b x \right) \right] \Psi(t, x), \quad (\text{F.2})$$

subject to a Gaussian initial state

$$\Psi(0, x) = \left(\frac{1}{2\pi\sigma^2} \right)^{1/4} e^{-\frac{x^2}{4\sigma^2}}. \quad (\text{F.3})$$

Then, the expectation value of the position observable \hat{x} is given by

$$\langle \hat{x} \rangle_t = \frac{b}{a} (\cos(\sqrt{at}) - 1). \quad (\text{F.4})$$

Proof. Since the potential field is quadratic, we invoke the Ehrenfest theorem to obtain the following Hamilton's equation for $\langle \hat{x} \rangle_t$ and $\langle \hat{p} \rangle_t$:

$$\frac{d}{dt} \langle \hat{x} \rangle_t = \langle \hat{p} \rangle_t, \quad (\text{F.5})$$

$$\frac{d}{dt} \langle \hat{p} \rangle_t = -\langle f \rangle_t = -a \langle \hat{x} \rangle_t - b, \quad (\text{F.6})$$

where the initial data are given by

$$\langle \hat{x} \rangle_0 = \langle \hat{p} \rangle_0 = 0. \quad (\text{F.7})$$

Therefore, the closed-form formula of $\langle \hat{x} \rangle_t$ is obtained by solving the system of ODEs. \square

F.1.2 Fock space truncation method

We can rewrite the real-space Schrödinger operator using the position operator \hat{x} and the momentum operator $\hat{p} = -i\frac{\partial}{\partial x}$:

$$H = \frac{1}{2} \sum_{j=1}^d \hat{p}_j^2 + f(\hat{x}_1, \dots, \hat{x}_d). \quad (\text{F.8})$$

We define two operators:

$$\hat{a} = \frac{1}{\sqrt{2}}(i\hat{p} + \hat{x}), \quad \hat{a}^\dagger = \frac{1}{\sqrt{2}}(-i\hat{p} + \hat{x}). \quad (\text{F.9})$$

The operator \hat{a}^\dagger is the raising/creation operator, and \hat{a} is the lowering/annihilation operator. They are infinite-dimensional quantum operators with eigenstates being the number basis (corresponding to the eigenstates of quantum harmonic oscillators):

$$\hat{a}|0\rangle = 0, \quad \hat{a}|n\rangle = \sqrt{n}|n-1\rangle \quad (\text{for } n \geq 1), \quad (\text{F.10})$$

$$\hat{a}^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle \quad (\text{for } n \geq 0). \quad (\text{F.11})$$

We can express the creation and annihilation operators in the standard matrix form,

$$\hat{a}^\dagger = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ & \sqrt{2} & 0 & & \\ & & \sqrt{3} & 0 & \\ & & & \dots & \dots \end{bmatrix}, \quad \hat{a} = \begin{bmatrix} 0 & 1 & & & \\ & 0 & \sqrt{2} & & \\ & & 0 & \sqrt{3} & \\ & & & 0 & 2 \\ & & & & \dots & \dots \end{bmatrix}. \quad (\text{F.12})$$

Using the ladder operators, we can rewrite the position and momentum operators:

$$\hat{x} = \frac{1}{\sqrt{2}}(\hat{a}^\dagger + \hat{a}), \quad \hat{p} = \frac{i}{\sqrt{2}}(\hat{a}^\dagger - \hat{a}). \quad (\text{F.13})$$

In the matrix form, the two operators are infinite-dimensional tri-diagonal matrices.

$$\hat{x} = \frac{1}{\sqrt{2}} \sum_{j=0}^{\infty} \sqrt{j+1} (|j\rangle \langle j+1| + |j+1\rangle \langle j|), \quad (\text{F.14})$$

$$\hat{p} = \frac{i}{\sqrt{2}} \sum_{j=0}^{\infty} \sqrt{j+1} (-|j\rangle \langle j+1| + |j+1\rangle \langle j|). \quad (\text{F.15})$$

Similarly, we can compute the quadratic operators:

$$\hat{p}^2 = \frac{1}{2} \sum_{j=0}^{\infty} (2j+1) |j\rangle \langle j| - \frac{1}{2} \sum_{j=0}^{\infty} \sqrt{(j+1)(j+2)} (|j\rangle \langle j+2| + |j+2\rangle \langle j|), \quad (\text{F.16})$$

$$\hat{x}^2 = \frac{1}{2} \sum_{j=0}^{\infty} (2j+1) |j\rangle \langle j| + \frac{1}{2} \sum_{j=0}^{\infty} \sqrt{(j+1)(j+2)} (|j\rangle \langle j+2| + |j+2\rangle \langle j|), \quad (\text{F.17})$$

which are infinite-dimensional matrices with bandwidth 2.

To simulate the real-space quantum operator (F.8) on a quantum computer, we need to truncate the Fock space (spanned by the number basis) up to finite dimension N . Then, we

can construct Hamiltonian embeddings of the truncated quantum operator for real-machine implementation.

By truncating Fock space up to N levels, we have the truncated operators,

$$\hat{x} = \frac{1}{\sqrt{2}} \sum_{j=0}^{N-2} \sqrt{j+1} (|j\rangle \langle j+1| + |j+1\rangle \langle j|), \quad (\text{F.18})$$

$$\hat{p}^2 = \frac{1}{2} \sum_{j=0}^{N-1} (2j+1) |j\rangle \langle j| - \frac{1}{2} \sum_{j=0}^{N-3} \sqrt{(j+1)(j+2)} (|j\rangle \langle j+2| + |j+2\rangle \langle j|), \quad (\text{F.19})$$

$$\hat{x}^2 = \frac{1}{2} \sum_{j=0}^{N-1} (2j+1) |j\rangle \langle j| + \frac{1}{2} \sum_{j=0}^{N-3} \sqrt{(j+1)(j+2)} (|j\rangle \langle j+2| + |j+2\rangle \langle j|). \quad (\text{F.20})$$

Therefore, if we use the unary embedding, the encoding Hamiltonians are $(N-1)$ -qubit operators:

$$Q_{\hat{x}}^{\text{unary}} = \frac{1}{\sqrt{2}} \sum_{j=1}^{N-1} \sqrt{j} X_j, \quad (\text{F.21})$$

$$Q_{\hat{p}^2}^{\text{unary}} = \sum_{j=1}^{N-1} \hat{n}_j - \frac{1}{2} \sum_{j=1}^{N-2} \sqrt{j(j+1)} X_{j+1} X_j, \quad (\text{F.22})$$

$$Q_{\hat{x}^2}^{\text{unary}} = \sum_{j=1}^{N-1} \hat{n}_j + \frac{1}{2} \sum_{j=1}^{N-2} \sqrt{j(j+1)} X_{j+1} X_j. \quad (\text{F.23})$$

It is worth noting that we already discard the global phase in the embedding operators.

Similarly, if we choose the one-hot embedding, the encoding Hamiltonians are N -qubit operators:

$$Q_{\hat{x}}^{\text{one-hot}} = \frac{1}{\sqrt{2}} \sum_{j=1}^{N-1} \sqrt{j} X_{j+1} X_j, \quad (\text{F.24})$$

$$Q_{\hat{p}^2}^{\text{one-hot}} = \sum_{j=1}^N (j - \frac{1}{2}) \hat{n}_j - \frac{1}{2} \sum_{j=1}^{N-2} \sqrt{j(j+1)} \frac{(X_{j+2} X_j + Y_{j+2} Y_j)}{2}, \quad (\text{F.25})$$

$$Q_{\hat{x}^2}^{\text{one-hot}} = \sum_{j=1}^N (j - \frac{1}{2}) \hat{n}_j + \frac{1}{2} \sum_{j=1}^{N-2} \sqrt{j(j+1)} \frac{(X_{j+2} X_j + Y_{j+2} Y_j)}{2}. \quad (\text{F.26})$$

Again, the global phase is discarded.

With these embedding operators, we can use the rules in [Appendix A.2](#) to construct an embedding of the real-space Schrödinger operator with a quadratic potential field.

To simulate the N -level truncated Hamiltonian, we require a large penalty coefficient g . With a large g , the norm of the embedding Hamiltonian becomes so huge that the standard product formula requires a significant number of Trotter steps that is infeasible for near-term devices. To address this issue, we transform the embedding operators to the interaction picture and use the quantum simulation algorithm qDRIFT (for details, see [Appendix A.3.3](#)).

F.1.3 Experiment setup and result

We perform experiments of the Schrödinger equation in one-dimensional real space using the IonQ Aria-1 processor. We use the penalty-free one-hot embedding to simulate the

real-space Schrödinger equation with quadratic potential $f(x) = x^2 - \frac{1}{2}x$. The embedding Hamiltonian is then given by

$$H_{\text{real-space}} = \frac{1}{2}Q_{\hat{p}^2}^{\text{one-hot}} - \frac{1}{2}Q_{\hat{x}^2}^{\text{one-hot}} + \frac{1}{2}Q_{\hat{x}}^{\text{one-hot}}, \quad (\text{F.27})$$

as described in [Appendix F.1.2](#).

We truncate the Hamiltonian to $N = 5$ levels, corresponding to 5 qubits using the one-hot embedding. The initial state is the first encoded state of the one-hot code, prepared using a single X rotation gate. We perform quantum simulation using the randomized first-order Trotter-Suzuki formula for evolution times up to $T = 5$ with $r = 11$ Trotter steps. For each circuit, the total gate count is 154 two-qubit gates in addition to a single one-qubit gate for preparing the initial state. To compute the expected position observable \hat{x} , we measure in x -basis using 1500 shots. For computing the expected kinetic energy $\frac{1}{2}\hat{p}^2$, we also measure in the computational basis using 500 shots.

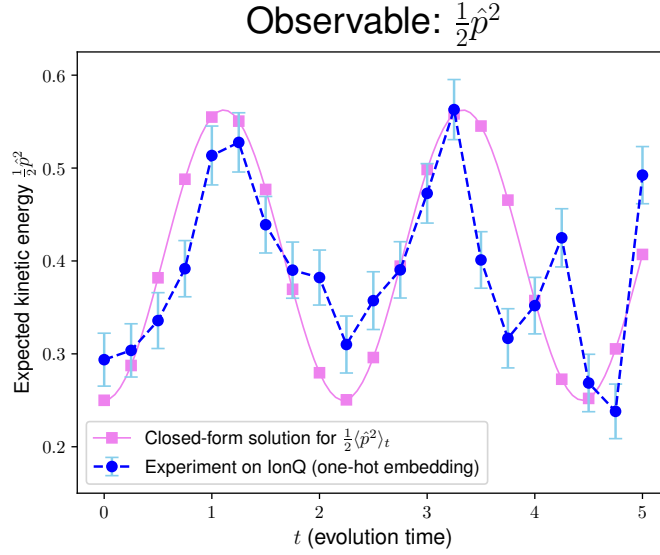


Figure 9: Expectation value of kinetic energy $\frac{1}{2}\hat{p}^2$ depicted as a function of evolution time t . The violet curve denotes the analytically computed expected kinetic energy with squares drawn at the same evolution times as the experiment data for comparison. Blue dots denote experimental results obtained from the IonQ Aria-1 processor using 1500 shots for x -basis measurements and 500 shots for z -basis measurements for each data point. Error bars denote standard error.

In [Figure 4C](#), we plot the expected value of the position observable. In violet, we plot the closed form solution, given by $\langle \hat{x} \rangle_t = \frac{1}{4}(1 - \cos(\sqrt{2}t))$. In blue, we plot the real-machine data obtained from the IonQ processor. The experimentally observed values show a sinusoidal trend consistent with the analytically predicted values.

In addition, we present the expected kinetic energy $\frac{1}{2}\langle \hat{p}^2 \rangle$ in [Figure 9](#). The analytical solution, shown in violet, is evaluated as $\frac{1}{2}\langle \hat{p}^2 \rangle = -\frac{5}{16} \cos^2(\sqrt{2}t) + \frac{9}{16}$. The experimental data is depicted by blue dots, showing a sinusoidal trend consistent with the analytically predicted solution.

F.1.4 Resource analysis

We conduct a resource analysis for performing a simulation of the real-space Schrödinger equation via the truncated Fock space method. The task is to perform quantum simulation

for a fixed evolution time T . To be consistent with the real-machine experiment, we fix the evolution time to be $T = 5$ and the potential function is $f(x) = \frac{1}{2}ax^2 + bx$ with $a = 2$ and $b = -1/2$. For this task, we apply the randomized first-order Trotter-Suzuki product formula.

For the standard binary encoding, we start with the desired quantum operator with $2^{\lceil \lg N \rceil}$ levels, in order to preserve the structure of the problem. To simulate the operator with N levels, we only use the first N basis states as the encoding subspace. To this end, we remove all matrix entries that contribute to interactions between the encoding subspace and the orthogonal complement. We use Qiskit to compute the Pauli decomposition of the resulting matrix and convert the circuit to single-qubit Pauli rotations and XX -rotation gates.

In [Figure 4B](#), we plot estimates of the total gate count required for the standard binary code and the penalty-free one-hot embedding. For standard binary, we observe an asymptotic gate complexity of $O(n^{3.16})$. For the one-hot embedding, the complexity is estimated as $O(n^{2.67})$, showing a modest advantage in the asymptotic scaling. In addition, we observe an order of magnitude advantage in the constant factor for the one-hot embedding.

F.2 Experiments on QuEra

F.2.1 Finite difference method

We now discuss how to simulate two-dimensional Schrödinger equations on the QuEra machine. Consider the following partial differential equation defined on the unit square $\Omega = [0, 1]^2$,

$$i \frac{\partial}{\partial t} \Psi(t, \mathbf{x}) = \left(-\frac{1}{2} \nabla^2 + V(\mathbf{x}) \right) \Psi(t, \mathbf{x}), \quad (\text{F.28})$$

where $\Psi(t, \mathbf{x}): [0, t] \times \Omega \rightarrow \mathbb{C}$ is the quantum wave function with Dirichlet boundary conditions (i.e., $\Psi(t, \mathbf{x}) = 0$ for $\mathbf{x} \in \partial\Omega$). $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator, and $V(\mathbf{x}): \Omega \rightarrow \mathbb{R}$ is the potential field.

In our QuEra experiment, we simulate the Schrödinger equation [\(F.28\)](#) using the finite difference method. Suppose that we discretize the unit interval $[0, 1]$ into N pieces with grid points $\{z_k = k/(N-1): k = 0, \dots, N-1\}$. For any function $u(x)$ defined on $[0, 1]$ with vanishing boundary condition $u(0) = u(1) = 0$, we can discretize u on the mesh $\{z_k\}$ and represent it as an N -dimensional vector:

$$\vec{u} = [u(z_0), \dots, u(z_{N-1})]^\top.$$

Using the central finite difference approximation, i.e., $u''(x) \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$, we can approximate the function u'' by a new N -dimensional vector $D\vec{u}$ (with $h = 1/(N-1)$), where the matrix D is given by

$$D = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ \dots & \dots & \dots & \dots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}. \quad (\text{F.29})$$

The matrix D is often called the finite difference discretization of the differential operator $\frac{\partial^2}{\partial x^2}$. Note that the matrix D is tri-diagonal. We can use either unary embedding or antiferromagnetic embedding for the matrix D (see [Appendix B.1](#)).

Similarly, the finite difference method can be applied to the 2D Schrödinger equation (F.28). The discretized quantum Hamiltonian reads,

$$\hat{H} = -\frac{1}{2}(D \otimes I) + I \otimes D + U, \quad (\text{F.30})$$

where I is the N -by- N identity matrix and U is a N^2 -by- N^2 diagonal matrix such that

$$U |z_j\rangle |z_k\rangle = V(z_j, z_k) |z_j\rangle |z_k\rangle.$$

F.2.2 Neutral-atom simulator and antiferromagnetic embedding

The QuEra quantum simulator is based on neutral atoms. The atoms are placed individually and deterministically on a two-dimensional plane by optical tweezers [108, 109]. On the QuEra quantum simulator, a qubit is realized by an atom with an internal ground state $|0\rangle$ and an excited Rydberg state $|1\rangle$. The Rydberg atoms are long-lived and can be coherently controlled by external laser pulses. The evolution of the neutral atoms is described by the Schrödinger equation $i\hbar \frac{d}{dt} |\psi(t)\rangle = \hat{H}(t) |\psi(t)\rangle$, where the system Hamiltonian is given by

$$\frac{\hat{H}(t)}{\hbar} = \sum_j \left(\frac{\Omega_j(t)}{2} \hat{X}_j - \Delta_j(t) \hat{n}_j \right) + \sum_{j < k} \frac{C_6}{|\mathbf{r}_j - \mathbf{r}_k|^6} \hat{n}_j \hat{n}_k. \quad (\text{F.31})$$

In the Hamiltonian, $\hat{X}_j = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\hat{n}_j = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ are the Pauli- X operator and the number operator acting on the j -th qubit, respectively. $\Omega_j(t)$, $\Delta_j(t)$ denotes the Rabi frequency and local detuning of the driving laser field on qubit j . C_6 is the Rydberg interaction constant that depends on the particular Rydberg atom used. \mathbf{r}_j denotes the position vector of the j -th qubit. In our experiment, we use the neutral-atom analog quantum computer fabricated by QuEra Computing Inc. In each simulation, we can initiate no more than 256 atoms, and the Rydberg interaction constant is $C_6 = 862690 \times 2\pi$ MHz $\cdot \mu\text{m}^6$.

In the system Hamiltonian (F.31), we have the atom-atom interactions $\hat{n}_j \hat{n}_k$ with positive coefficients. This structure is very suitable for antiferromagnetic embedding. In Lemma 19, we show how to engineer a penalty Hamiltonian for antiferromagnetic embedding by tuning the local detuning of neutral atoms.

Lemma 19. *Given an integer $N > 1$, let $n = N - 1$. Define the penalty Hamiltonian:*

$$\hat{H}_{\text{pen}} = \sum_{1 \leq j < k \leq n} \frac{C_6}{|\mathbf{r}_j - \mathbf{r}_k|^6} \hat{n}_j \hat{n}_k - \sum_{j=1}^n \Delta_j \hat{n}_j, \quad (\text{F.32})$$

where the atoms are placed on a straight line with equal distance $r > 0$, i.e., $\mathbf{r}_j = (0, (j-1)r)$, and the local detuning values are:

$$\Delta_j = \begin{cases} \frac{C_6}{r^6} \left(\sum_{m=1}^{\lfloor (n+1-j)/2 \rfloor} \frac{1}{(2m-1)^6} + \sum_{m=1}^{\lfloor j/2 \rfloor} \frac{1}{(2m)^6} \right), & (\text{for odd } j) \\ \frac{C_6}{r^6} \left(\sum_{m=1}^{\lfloor (n-j)/2 \rfloor} \frac{1}{(2m-1)^6} + \sum_{m=1}^{\lfloor (j-1)/2 \rfloor} \frac{1}{(2m)^6} \right). & (\text{for even } j) \end{cases} \quad (\text{F.33})$$

Then, the penalty Hamiltonian (F.32) has an N -fold degenerate ground-energy subspace \mathcal{S} which is spanned by all the n -bit antiferromagnetic code (see Table 8).

Proof. We use $|w_k\rangle$ to denote the antiferromagnetic code for $k = 1, \dots, N$. Define the following two operators

$$H'_Z = \sum_{1 \leq j < k \leq n} \frac{C_6}{|\mathbf{r}_j - \mathbf{r}_k|^6} \hat{n}_j \hat{n}_k, \quad H''_Z = - \sum_{j=1}^n \Delta_j \hat{n}_j,$$

where $\mathbf{r}_j = (0, (j-1)r)$, and we consider an ansatz

$$\hat{H}_{\text{pen}} = H'_Z + H''_Z.$$

Since the Hamiltonian H'_Z is in antiferromagnetic ordering, its low-energy subspace is spanned by antiferromagnetic code. However, these low-energy states are not degenerate. To get the N -degenerate ground-energy subspace, we solve a linear system to compute Δ_j in \hat{H}_{pen} .

First, we try to match the energy of $|w_1\rangle$ and $|w_2\rangle$. This requires the following identity holds:

$$\langle w_1 | \hat{H}_{\text{pen}} | w_1 \rangle = \langle w_2 | \hat{H}_{\text{pen}} | w_2 \rangle. \quad (\text{F.34})$$

Note that $w_1 = 0101\dots$ and $w_2 = 1101\dots$, and the 2-tensor $\hat{n}_j \hat{n}_k = |11\rangle_{jk} \langle 11|_{jk}$ is visible only when the atoms j and k are simultaneously excited, so we have that,

$$\langle w_2 | H'_Z | w_2 \rangle - \langle w_1 | H'_Z | w_1 \rangle = \sum_{m=1}^{\lfloor n/2 \rfloor} \frac{C_6}{r^6 (2m-1)^6}.$$

Meanwhile, we have $\langle w_2 | H''_Z | w_2 \rangle - \langle w_1 | H''_Z | w_1 \rangle = -\Delta_1$. To have the identity (F.34), we must have

$$\Delta_1 = \frac{C_6}{r^6} \sum_{m=1}^{\lfloor n/2 \rfloor} \frac{1}{(2m-1)^6}.$$

Similarly, we can compute Δ_2 . Comparing $w_2 = 1101\dots$ and $w_3 = 1001\dots$, we have

$$\langle w_2 | H_Z^{(1)} | w_2 \rangle - \langle w_3 | H_Z^{(1)} | w_3 \rangle = \frac{C_6}{r^6} + \sum_{m=1}^{\lfloor n/2 \rfloor - 1} \frac{C_6}{r^6 (2m)^6}.$$

Also, we have $\langle w_2 | H''_Z | w_2 \rangle - \langle w_3 | H''_Z | w_3 \rangle = -\Delta_2$, it follows that

$$\Delta_2 = \frac{C_6}{r^6} \left(\frac{1}{1^6} + \sum_{m=1}^{\lfloor n/2 \rfloor - 1} \frac{1}{(2m)^6} \right).$$

We can compute all local detuning Δ_j following the same procedure. It turns out that if we choose Δ_j as in (F.33), the ground-energy subspace of \hat{H}_{pen} is indeed spanned by the codeword states $|w_j\rangle$ for $j = 1, \dots, N$. \square

F.2.3 Experiment setup and result

We perform experiments of simulating the 2D Schrödinger equation using the QuEra Aquila processor. The programmable quantum Hamiltonian on QuEra Aquila reads [110],

$$\frac{\hat{H}(t)}{\hbar} = \frac{\Omega(t)}{2} \left(\sum_j e^{i\phi(t)} |0\rangle \langle 1|_j + e^{-i\phi(t)} |1\rangle \langle 0|_j \right) - \Delta(t) \left(\sum_j \hat{n}_j \right) + \sum_{j < k} \frac{C_6}{|\mathbf{r}_j - \mathbf{r}_k|^6} \hat{n}_j \hat{n}_k, \quad (\text{F.35})$$

where $\Omega(t)$, $\phi(t)$, and $\Delta(t)$ denote the Rabi frequency, laser phase, and the detuning of the driving laser field on atom (qubit) j coupling the two states $|0\rangle$ (ground state) and $|1\rangle$ (Rydberg state). It is worth noting that all programmable control parameters are *global*.

Hamiltonian embedding and the engineering of the potential field. We use the antiferromagnetic embedding to simulate the quantum dynamics. Ideally, we want to use [Lemma 19](#) to engineer a perfect antiferromagnetic embedding of the discretized Laplacian operator $-\frac{1}{2}(D \otimes I + I \otimes D)$ (see [\(F.30\)](#)). However, currently (as of the fall of 2023), the QuEra Aquila processor does not support user-specified local detuning (see [\(F.35\)](#)). The lack of local detuning poses a significant restriction on the Hamiltonian we can engineer on the QuEra Aquila processor.

In our QuEra experiment, we follow the intuition of [Lemma 19](#) and arrange 12 neutral atoms as in [Figure 4D](#). The 12 neutral atoms are grouped into 2 vertical chains. Each chain has 6 atoms. The (sub-)system Hamiltonian for each chain reads

$$H_0 = \sum_{1 \leq j < k \leq 6} \frac{C_6}{(k-j)^6 r^6} \hat{n}_j \hat{n}_k - \Delta \sum_{j=1}^6 \hat{n}_j, \quad (\text{F.36})$$

where $r = 5.9 \mu\text{m}$ is the distance between two adjacent atoms, and Δ is a global detuning.

It is clear that the actual Hamiltonian [\(F.36\)](#) is similar to the penalty Hamiltonian specified in [\(F.32\)](#), except that we can not specify individual detuning Δ_j for each atom. In this case, the codeword subspace of antiferromagnetic embedding

$$\mathcal{S} = \{|010101\rangle, |010100\rangle, |010110\rangle, |010010\rangle, |011010\rangle, |001010\rangle, |101010\rangle\}$$

coincide with the low-energy subspace of H_0 but not exactly the ground-energy subspace. This non-degeneracy can be realized as an *effective* potential field. Namely, we may regard the operator [\(F.32\)](#) as the *desired* penalty Hamiltonian (as in [\(F.32\)](#)) plus an operator that embeds a potential field (although this potential field is not very smooth and it does not correspond to a familiar closed-form function). Similarly, the system Hamiltonian of the two atom chains ([Figure 4D](#)) can be regarded as the penalty Hamiltonian for 2D antiferromagnetic embedding plus an effective potential operator. In [Figure 10A](#), we plot the effective 2D potential function at $\Delta = 5 \times 10^7$ rad/s. The potential is shown as a discretized function on the 2D mesh $\{(z_j, z_k) : j, k = 1, \dots, N\}$.

Initial state preparation. The QuEra Aquila machine is initialized to the all-zeros state. This default initial state is not in the embedding subspace of the antiferromagnetic embedding. In our experiment, we use a *3-stage* state preparation subroutine [\[31\]](#) to prepare an initial state that has a significant overlap (around 73%) with the embedding subspace. Since the initial all-zeros state is the ground state when $\Omega = 0$ MHz and $\Delta < 0$ MHz, this state preparation pulse is quasi-adiabatic and prepares a state with high overlap with the antiferromagnetic encoding subspace. In our pulse, we use a trapezoidal pulse for the Rabi frequency with maximum frequency $\Omega_{\text{max}} = 15.8$ MHz and a linear ramp for the detuning from -50 MHz to 50 MHz. The duration of initial state preparation is $0.5 \mu\text{s}$. The initial state preparation pulses are shown in [Figure 10C](#) (see the first $0.5 \mu\text{s}$). The engineered initial state (e.g., the quantum state at $t = 0.5 \mu\text{s}$) is shown in [Figure 10B](#).

Time-rescaling and parameters for analog quantum simulation. On the QuEra Aquila device, all parameters are specified with a physical unit. We conducted our experiments through Amazon Braket, where the unit of the Rabi frequency Ω is radians per second (rad/s) and the evolution is in the unit of second (s) [\[64, Section 1.3\]](#). Note that $1 \text{ rad/s} = \frac{1}{2\pi} \text{ Hz}$.

We can establish a relation between these QuEra physical parameters with the abstract unitless model Hamiltonian [\(F.30\)](#) via time-rescaling. We assume the quantum

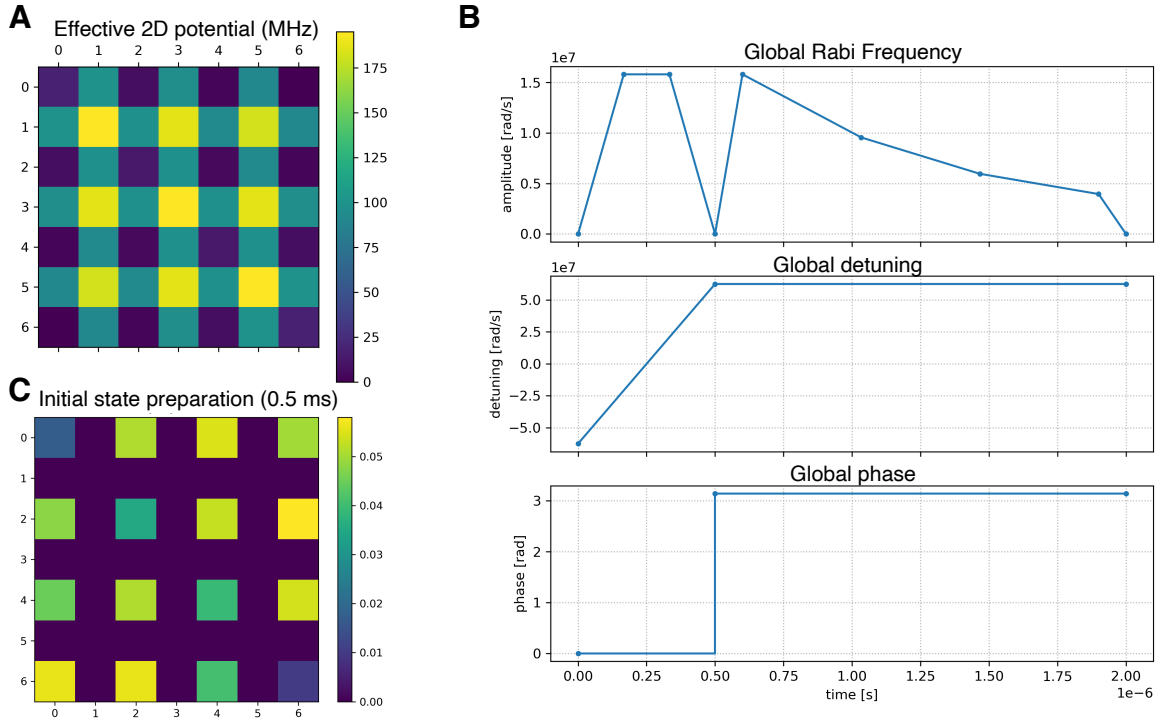


Figure 10: Simulating real-space dynamics on QuEra Aquila. **A.** The effective potential field engineered by Rydberg interactions. The unit of the potential is MHz. The potential field is normalized such that the minimal potential is 0.

simulation is restricted to the unit square $[0, 1]^2$ (with Dirichlet boundary conditions), and each dimension is discretized into 8 pieces. Therefore, the discretized differential operator is D in (F.29) with $h = 1/8$. Therefore, simulating a physical Hamiltonian (on QuEra Aquila) $Q = -\frac{\Omega(t)}{2} \sum_j X_j$ for time T (seconds) is equivalent to simulating the Hamiltonian $Q = -\varphi(t) \frac{1}{2h^2} \sum_j X_j$ (i.e., the antiferromagnetic embedding of the discretized Laplacian $\varphi(t)D/2h^2$) for (unitless) time

$$T_{\text{eff}} = 10^6 \cdot T,$$

where $\varphi(t) = \frac{h^2 \Omega(t)}{2\pi \cdot 10^6}$. In our experiments, we run the simulation on QuEra Aquila for $T = 0.5, 1, 1.5 \mu\text{s}$ (plus an initial state preparation duration of $0.5 \mu\text{s}$). Effectively, we are simulating the real-space quantum dynamics for $T_{\text{eff}} = 0.5, 1, 1.5$.

Post-selection. The global Rabi frequency corresponds to the time-dependent function (i.e., $\varphi(t)$) in the kinetic energy. We gradually decrease the global Rabi frequency (see Figure 10B for the control pulses with total evolution time $T = 2.0 \mu\text{s}$) so the quantum state will eventually find the low-energy configuration regarding the effective potential field. We consider a measured sample *legit* if it is in the embedding subspace. In our experiment on QuEra Aquila, approximately half of the samples in the measurement are legit. In Figure 4E, we plot the distribution of the legit samples for evolution time $T = 0.5, 1, 1.5$ (corresponding to total physical evolution time $T = 1, 1.5, 2 \mu\text{s}$).