Commun. Math. Phys. (2026) 407:16
Digital Object Identifier (DOI) https://doi.org/10.1007/s00220-025-05517-w

Communications in
Mathematical
Physics

Testing and Learning Structured Quantum Hamiltonians

Srinivasan Arunachalam[1], **Arkopal Dutt**[2], **Francisco Escudero Gutiérrez**[3]

[1] IBM Quantum, Almaden Research Center, San Jose, CA, USA. E-mail: Srinivasan.Arunachalam@ibm.com
[2] IBM Quantum, Cambridge, MA, USA. E-mail: arkopal@ibm.com
[3] Qusoft and CWI, Amsterdam, The Netherlands. E-mail: feg@cwi.nl

Received: 9 June 2025 / Accepted: 6 November 2025

Abstract: We consider the problems of testing and learning an n-qubit Hamiltonian $H = \sum_x \lambda_x \sigma_x$ expressed in its Pauli basis, from queries to its evolution operator $U = e^{-iHt}$. To this end, we prove the following results.

1. **Testing**: We give a *tolerant* testing protocol to decide if a Hamiltonian is ε_1-close to k-local or ε_2-far from k-local in the ℓ_2 norm of the coefficients, with $O(1/(\varepsilon_2-\varepsilon_1)^4)$ queries, thereby solving two open questions posed in a recent work by Bluhm, Caro and Oufkir (Bluhm, A., Caro, M.C., Oufkir, A.). We give a protocol for testing whether a Hamiltonian is ε_1-close to being s-sparse or ε_2-far from being s-sparse in the ℓ_2 norm of the coefficients, with $O(s^6/(\varepsilon_2^2-\varepsilon_1^2)^6)$ queries.
2. **Learning**: We give a protocol to ε-learn unstructured Hamiltonian in the ℓ_∞ norm of the coefficients with $O(1/\varepsilon^4)$ queries. Combining this with the non-commutative Bohnenblust-Hille inequality, we obtain an algorithm for learning k-local Hamiltonians in ℓ_2 norm of the coefficients that only uses $O(\exp(k^2 + k\log(1/\varepsilon)))$ queries. For Hamiltonians that are s-sparse in the Pauli basis, we can learn them in the ℓ_2 norm with $O(s^2/\varepsilon^4)$ queries.
3. **Learning without quantum memory**: The learning results stated above have no dependence on the system size n, but require n-qubit quantum memory. We give subroutines that allow us to reproduce all the above learning results without quantum memory; squaring the query complexity and paying a $(\log n)$-factor in the local case and an n-factor in the sparse case.
4. **Testing without quantum memory**: We give a new subroutine called *Pauli hashing*, which allows one to tolerantly test s-sparse Hamiltonians in the ℓ_2 norm using $\tilde{O}(s^{14}/(\varepsilon_2^2-\varepsilon_1^2)^{18})$ query complexity. A key ingredient is showing that s-sparse Pauli channels can be tested in a tolerant fashion as being ε_1-close to being s-sparse or ε_2-far under the diamond norm, using $\tilde{O}(s^2/(\varepsilon_2-\varepsilon_1)^6)$ queries via Pauli hashing.

In order to prove these results, we prove new structural theorems for local Hamiltonians, sparse Pauli channels and sparse Hamiltonians. We complement our learning algorithms

Published online: 08 December 2025

with lower bounds that are polynomially weaker. Furthermore, our algorithms use short time evolutions and do not assume prior knowledge of the terms on which the Pauli spectrum is supported on, i.e., we do not require prior knowledge of the *support* of the Hamiltonian.

Contents

1. Introduction

A fundamental and significant challenge with building quantum devices is being able to characterize and calibrate its behavior. One approach to do so is *Hamiltonian learning* which seeks to learn the Hamiltonian governing the dynamics of a quantum system given finite classical and quantum resources. Beyond system characterization, it is also carried out during validation of physical systems and designing control strategies for implementing quantum gates [IBF+20]. However, learning an n-qubit Hamiltonian is known to be difficult, requiring complexity that scales exponential in the number of qubits unless a coarse metric is used [Car23].

In practice, prior knowledge on the structure of Hamiltonians is available e.g., those of engineered quantum devices [SMCG16] where the underlying Hamiltonians primarily

involve local interactions with few non-local interactions, and even naturally occurring physical quantum systems such as those with translationally invariant Hamiltonians. To highlight these structural properties, consider an n-qubit Hamiltonian H (which is a self-adjoint operator acting on $(\mathbb{C}^2)^{\otimes n}$) expanded in terms of the n-qubit Pauli operators:

$$H = \sum_{x \in \{0,1\}^{2n}} \lambda_x \sigma_x,$$

where λ_x are real-valued coefficients (also called interaction strengths) of the Pauli operators σ_x denoted by the string $x = (a, b) \in \mathbb{F}_2^{2n}$ with $\sigma_{(a,b)} = i^{a \cdot b} \otimes_{i=1}^{n} X^{a_i} Z^{b_i}$. We call the set of Paulis with non-zero coefficients λ_x as the Pauli spectrum of the Hamiltonian denoted by $\mathcal{S} = \{x \in \{0,1\}^{2n} : \lambda_x \neq 0\}$. Of particular relevance are *k-local* Hamiltonians which involve Pauli operators that act non-trivially on at most k qubits and *s-sparse* Hamiltonians whose Pauli expansion contains at most s non-zero Pauli operators i.e., $|\mathcal{S}| \leq s$.

There has thus been a growing suite of Hamiltonian learning results that have shown that when the underlying n-qubit Hamiltonian H satisfies these structural properties, learning can be performed with only poly(n) query complexity, either by making "queries" to the unitary evolution operator $U(t) = \exp(-iHt)$ [dSLCP11,HBCP15,ZYLB21, HKT22,YSHY23,DOS23,HTFS23,LTN+23,SFMD+24,GCC24,Zha24,HMG+25], or by assuming one has access to Gibbs state [AAKS21,HKT22,RSF23,ORSFW23 BLMT23,GCC24]. Notably, [BLMT24] considered the problem of learning Hamiltonians that are both local and sparse, without prior knowledge of the support. Several of the learning algorithms mentioned above however require assumptions on the support of the Hamiltonian beyond locality or sparsity, such as [HTFS23] which considers *geometrically-local* Hamiltonians (a subset of local Hamiltonians) and [YSHY23] which requires assumptions on the support.

Moreover, before learning, it might be desirable to uncover *what is the structure* of an unknown Hamiltonian in order to choose specialized learning algorithms. Even deciding if a Hamiltonian has a particular structure is a fundamental challenge and constitutes the problem of *testing* if an unknown Hamiltonian satisfies a certain structural property. As far as we know, this line of investigation is nascent with only a few works on Hamiltonian *property* testing [SY23,ACQ22,LW22] with Blum et al. [BCO24b] having considered the problem of testing local Hamiltonians and the problem of testing sparse Hamiltonians yet to be tackled. This leads us to the motivating question of our work:

What is the query complexity of learning and testing structured Hamiltonians?

1.1. Problem statement. Before we state our results answering the question above, we clearly mention our learning and testing problems first. If H is the Hamiltonian describing the dynamics of a certain physical system, then the state of that system evolves according to the *time evolution operator* $U(t) = e^{-iHt}$. This means that if $\rho(0)$ is the state at time 0, at time t the state would have evolved to $\rho(t) = U(t)\rho(0)U^\dagger(t)$. Hence, to test and learn a Hamiltonian one can do the following: prepare a desired state, apply $U(t)$ or tensor products of $U(t)$ with identity to the state, and finally measure in a chosen basis. From here onwards, this is what we mean by *querying* the unitary $U(t)$. It is usual to impose the normalization condition $\|H\|_{\text{op}} \leq 1$ (i.e., that the eigenvalues of H are

bounded in absolute value by 1). We will assume this normalization unless otherwise stated, but we will also work out the dependence on $\|H\|_{\text{op}}$ for our learning algorithms. Throughout this paper, we will consider the normalized Frobenius norm as the distance between Hamiltonians, unless otherwise stated. This distance is

$$d(H, H') = \|H - H'\|_2 = \sqrt{\frac{\text{Tr}[(H - H')^2]}{2^n}},$$

and it equals the ℓ_2-norm of the Pauli spectrum, $d(H, H') = \sqrt{\sum |\lambda_x - \lambda'_x|^2}$. A *property* of a Hamiltonian, denoted $\mathcal{H}$ is a class of Hamiltonians that satisfy the property (here we will be interested in sparse and local properties). We say that H is ε-far from having a property $\mathcal{H}$ if $d(H, H') > \varepsilon$ for every $H' \in \mathcal{H}$, and otherwise is ε-close. Now, we are ready to state the testing and learning problems.

Let $\mathcal{H}$ be a property and let H be an unknown Hamiltonian $\text{Tr}[H] = 0$.

Problem 1.1. (Tolerant testing) Promised H is either ε_1-close to or ε_2-far from satisfying property $\mathcal{H}$, decide which is the case by making queries to $U(t)$.

Problem 1.2. (Hamiltonian learning) Promised $H \in \mathcal{H}$, output a classical description of $\widetilde{H} \in \mathcal{H}$ such that $\|H - \widetilde{H}\|_2 \leq \varepsilon$ by making queries to $U(t)$.

1.2. Summary of results. The main results of this work are query-efficient algorithms for testing and learning Hamiltonians that are local and/or sparse. We can reproduce these results without using quantum memory by increasing the number of queries. For simplicity, in the introduction we assume that $\|H\|_{\text{op}} = O(1)$, but in the main text we make explicit the dependence of our algorithms with $\|H\|_{\text{op}}$. We summarize our results with constant accuracy (constant parameters $\varepsilon, \varepsilon_1, \varepsilon_2$) in the Table 1.

Before we discuss our results in more detail, we make a few remarks about our main results.

(*i*) As far as we know, this is the first work: (*a*) with complexities that are *independent* of n (with memory)[1], and (*b*) that does not assume knowledge of the support.[2]

(*ii*) We give the first learning algorithm for Hamiltonians that are only promised to be sparse, and not necessarily local. Similarly, our local Hamiltonian learning problem doesn't assume geometric locality which was assumed in several prior works.

(*iii*) Our testing algorithms are tolerant, i.e., they can handle the setting where $\varepsilon_1 \neq 0$. As far as we know, there are only a handful of polynomial-time tolerant testers for quantum objects.

[1] There are a few works that achieve n-independent complexities for learning local Hamiltonians in the ∞-norm of the Pauli coefficients, but when transformed into 2-norm learners they yield complexities depending on n^k.

[2] Soon after the third-author's work [Gut24], Bakshi et al. [BLMT24] presented a learning algorithm that does not require prior knowledge of the support, achieving Heisenberg scaling using heavy machinery.

Table 1. Total time evolution for learning and testing n-qubit structured Hamiltonians

	Testing		**Learning**	
	with memory	w/o memory	with memory	w/o memory
s-sparse	$O(s^{1.5})$ Theorem 4.4	$O(s^{14})$ Theorem 5.7	$O(s^3)$ Theorem 4.9	$O(ns^3)$ Theorem 5.1
k-local	$O(1)$ Theorem 4.1	$O(1)$ [BCO24b]	$\exp(k^2)$ Theorem 4.7	$(\log n)\cdot\exp(k^2)$ Theorem 5.1
k-local & s-sparse	$O(s^{1.5})$	$O(s^{14})$	$\min\{\exp(k^2), O(s^3)\}$	$k\log(n)\cdot\min\{\exp(k^2), O(s^3)\}$

The query complexities are only polynomially higher. Dependence on n and the structural property is shown for constant accuracy. Results are indicated with quantum memory (i.e., an n-qubit ancillary system is available) and without quantum memory

(*iv*) We show that all our learning protocols with quantum memory can be translated to ones which require no quantum memory. In the case of learning structured Hamiltonians, we obtain a protocol with only a factor $\log n$ overhead for local Hamiltonians and a protocol with a factor n overhead for sparse Hamiltonians.[3]

(*v*) We also give a tolerant testing algorithm for s-sparse Hamiltonians that requires no quantum memory based on a new subroutine called Pauli hashing. The query complexity is $\text{poly}(s)$ and is notably independent of dimension n.

(*vi*) Our learning algorithms are based on a subroutine that estimates arbitrary n-qubit Hamilotmians with $O(1/\varepsilon^4)$ queries, albeit in the coarser metric of the ℓ_∞-norm of the Pauli coefficients. As far as we know, this is the best result for unstructured Hamiltonians. Notably, it is also the first time-efficient proposal for this problem.

We remark that most previous work on Hamiltonian learning (that we highlighted earlier) are done under the distance induced by the supremum norm of the Pauli spectrum and with extra constraints apart from locality [dSLCP11,HBCP15,ZYLB21,HKT22,WKR+22,YSHY23,Car23,DOS23,HTFS23,LTN+23,MBC+23,SFMD+24,GCC24]. When transformed into learning algorithms under the finer distance induced by the ℓ_2-norm of the Pauli spectrum, these proposals yield complexities that depend polynomially on n^k and only work for a restricted family of k-local Hamiltonians. The works that explicitly consider the problem of learning under the ℓ_2-norm have complexities depending on n and assume a stronger access model [CW23,BLMT24].

1.3. Results. **Testing.** Recently, Bluhm, Caro and Oufkir proposed a non-tolerant testing algorithm, meaning that it only works for the case $\varepsilon_1 = 0$, whose query complexity is $O(n^{2k+2}/(\varepsilon_2 - \varepsilon_1)^4)$ and with total evolution time $O(n^{k+1}/(\varepsilon_2 - \varepsilon_1)^3)$. They posed as open questions whether the dependence on n could be removed and whether an efficient tolerant-tester was possible [BCO24a, Section 1.5]. Our first result gives positive answer to both questions.

Result 1.3. *There is an algorithm that solves Problem 1.1 for k-local Hamiltonians by making* $\text{poly}(1/(\varepsilon_2 - \varepsilon_1))$ *queries to the evolution operator and with* $\text{poly}(1/(\varepsilon_2 - \varepsilon_1))$ *total evolution time.*

See Theorem 4.1 for a formal statement of this result. Our algorithm to test for locality is simple. It consists of repeating the following process $1/(\varepsilon_2 - \varepsilon_1)^4$ times: prepare n EPR pairs, apply $U(\varepsilon_2 - \varepsilon_1) \otimes \text{Id}_{2^n}$ to them and measure in the Bell basis. Each time that we repeat this process, we sample from the Pauli sprectrum of $U(\varepsilon_2 - \varepsilon_1)$.[4] As $\varepsilon_2 - \varepsilon_1$ is small, Taylor expansion ensures that $U(\varepsilon_2 - \varepsilon_1) \approx \text{Id}_{2^n} - i(\varepsilon_2 - \varepsilon_1)H$, so sampling from the Pauli spectrum of $U(\varepsilon_2 - \varepsilon_1)$ allows us to estimate the weight of the non-local terms of H. If that weight is big, we output that H is far from k-local, and otherwise we conclude that H is close to k-local.

Classically there have been many papers studying the problem of testing and learning sparse Boolean functions [GOS+11,NS12,YZ20,EIS22], however there are not many

[3] The difference in the overhead of n versus $\log(n)$ for learning Hamiltonians which are sparse versus local respectively, is due to the fact that in our memory-less learning algorithms, we estimate the squares of all the potentially non-zero Pauli coefficients. Upon applying an union bound, this incurs an overhead that scales logarithmically in the number of potentially non-zero Pauli coefficients. If the Hamiltonian is promised to be k-local, then there are $O(n^k)$ potential non-zero Pauli coefficients, whereas if the Hamiltonian is promised to be s-sparse, then there are 4^n potentially non-zero Pauli coefficients.

[4] The Pauli spectrum of a unitary $U = \sum_x \widehat{U}_x \sigma_x$ determines a probability distribution because $\sum_x |\widehat{U}_x|^2 = 1$.

results on learning *sparse Hamiltonians* (and not necessarily local). The only testing result that we are aware of requires $O(sn)$ queries and time $O(4^{ns})$ [BCO24b, Remark B.2]. Here, we present the first sparsity testing algorithm whose query complexity does not depend on n. In addition, it is the first time-efficient algorithm for this task.

Result 1.4. *There is an algorithm that solves Problem 1.1 for s-sparse Hamiltonians by making* $\mathrm{poly}(s/(\varepsilon_2-\varepsilon_1))$ *queries to the evolution operator and with* $\mathrm{poly}(s/(\varepsilon_2-\varepsilon_1))$ *total evolution time.*

See Theorem 4.3 for a formal statement. This testing algorithm consists on performing Pauli sampling of $U(\sqrt{(\varepsilon_2^2-\varepsilon_1^2)/s})$ a total of $O(s^4/(\varepsilon_2^2-\varepsilon_1^2)^4)$ times. From these samples one can estimate the sum of the squares of the top s Pauli coefficients of U. If this quantity is big enough, we output that the Hamiltonian is close to s-sparse, and otherwise that is far. Although from this high-level description the algorithm seems similar to the locality testing one, the analysis is more involved and requires taking the second order Taylor expansion, which is the reason why the dependence on $(\varepsilon_2-\varepsilon_1)$ is worse in this case.

Additionally, we provide a sparsity tester (Theorem 4.4) that only makes $O(s^2/\varepsilon_2^4)$ queries with $O(s^{1.5}/\varepsilon_2^3)$ total evolution time, but only works in the regime $\varepsilon_1 = O(\varepsilon_2/\sqrt{s})$.

Learning. We first propose a protocol to learn unstructured Hamiltonians efficiently in the coarser ℓ_∞ norm of the Pauli coefficients. Then, we turn it into a learner in the ℓ_2 norm for local and sparse Hamiltonians. In particular, we propose the first learning algorithm for sparse Hamiltonians which does not make any assumptions regarding the support of the Hamiltonian beyond sparsity.[5]

Result 1.5. *There is an algorithm that outputs estimates $\widetilde{\lambda}_x$ such that $|\lambda_x-\widetilde{\lambda}_x|\le\varepsilon$ for every $x\in\{0,1\}^{2n}$ by making $O(1/\varepsilon^4)$ queries to the evolution operator with $O(1/\varepsilon^3)$ total evolution time.*

See Theorem 4.6 for a formal result. The learning algorithm has two stages. In the first stage one samples from the Pauli distribution of $U(\varepsilon)$, as in the testing algorithm, and from that one can detect which are the big Pauli coefficients of H. In the second stage we learn the large Pauli coefficients via a subroutine based on Clifford Shadows (see Lemma 4.5). This subroutine allows to learn a set of m Pauli coefficients of a Hamiltonian with $\log(m)$ queries to its time evolution operator, which may be of independent interest. For Hamiltonians that are k-local, we have the following learning result in the ℓ_2-norm.

Result 1.6. *There is an algorithm that solves Problem 1.2 for k-local Hamiltonians by making* $\exp(k^2+k\log(1/\varepsilon))$ *queries to the evolution operator with* $\exp(k^2+k\log(1/\varepsilon))$ *total evolution time.*

See Theorem 4.7 for a formal statement of this result. In the case that the Hamiltonian is k-local, one can ensure that the coefficients not detected as big in the first stage of the algorithm of Result 1.5 have a neglectable contribution to the ℓ_2-norm, from which Result 1.6 follows. To argue this formally, we use the non-commutative Bohnenblust-Hille inequality, which has been used recently for various quantum learning algorithms [HCP23, VZ23]. For Hamiltonians that are s-sparse, we have the following learning result in the ℓ_2-norm.

[5] A concurrent work also dealt with the problem of learning sparse Hamiltonians [Zha24]. See Table 2 for a comparison.

Result 1.7. *There is an algorithm that solves Problem 1.2 for s-sparse Hamiltonians by making* $\mathrm{poly}(s/\varepsilon)$ *queries to the evolution operator with* $\mathrm{poly}(s/\varepsilon)$ *total evolution time.*

See Theorem 4.9 for a formal statement. Result 1.7 follows by adding a rounding step to the algorithm of Result 1.5 that ensures that all zero coefficients of the Hamiltonians are also zero for the approximating Hamiltonian.

Learning and testing without quantum memory. Motivated by the limitations of current devices, there has been a series of recent works to understand the power of quantum memory in testing and learning tasks, exhibiting exponential separations in some cases [CCHL22,CGY24,CLL24]. A natural question is, if the problems that we mentioned above become harder without quantum memory?

Learning without memory. We surprisingly show that, the learning protocols that we mention above, can be implemented efficiently when one has *no quantum memory*. To this end, we provide two crucial subroutines for (i) estimating the Pauli spectrum of a unitary, (ii) estimating a single Pauli coefficient to make our protocols work in the memory-less setting. Subroutine (ii) incurs in no extra query-cost, and subroutine (i) only incurs in a factor-n overhead in the case of learning s-sparse Hamiltonians and a factor $\log(n)$ in the case of learning k-local Hamiltonians. These subroutines can also be useful in other contexts. In particular, we give a tolerant tester that decides if an unknown n-qubit unitary is a k-junta using $O(4^k)$ queries (see Proposition 3.5), thereby making progress on a question of Chen et al. [CNY23, Section 1.3], and then we use subroutine (i) to turn it into a memory-less tester that makes only $O(4^k n)$ queries.

Testing sparse Pauli channels via Pauli hashing. In order to test the s-sparsity of Hamiltonians without memory we reduce the problem to that of testing the s-sparsity of a Pauli channel $\Phi : \rho \mapsto \sum_x p(x)\sigma_x \rho \sigma_x$, which is of independent interest. To tackle this latter problem, we introduce a new technique called *Pauli hashing*. Pauli hashing partitions the set of all n-qubit Pauli operators into $O(s)$ *buckets* depending on their commutation relations with a randomly chosen Pauli subgroup G. Each of the buckets contain all the Paulis that would have the same commutation relations with elements in G and these buckets in fact correspond to the cosets of the centralizer of G. This partitioning is useful for testing sparsity for two reasons, which we show: (i) one can efficiently compute the *energies* i.e., the sum of the error rates $p(x)$ falling into any given bucket; and (ii) the sum of the top s energies over the buckets is a good proxy of the sum of the top s error rates $p(x)$. Thus, one can estimate the sum of the top s error rates. Moreover, we show that this sum is close to 1 if and only if Φ is close to being s-sparse in the diamond norm, thereby allowing Pauli hashing to test for sparsity.

Result 1.8. *There is an algorithm with no quantum memory that tests if a Pauli channel is ε_1-close to or ε_2-far from being s-sparse in the diamond norm by making $\widetilde{O}(s^2/(\varepsilon_2 - \varepsilon_1)^6)$ queries to the channel.*

See Theorem 5.3 for a formal statement. We remark that Pauli hashing only requires the preparation of Pauli eigenstates and Pauli measurements, making it suitable for the near-term.

Testing sparse Hamiltonians without memory. We provide a memory-less testing algorithm for s-sparse Hamiltonians that uses Pauli hashing, that is completely independent of our tester with memory and only requires $\mathrm{poly}(s/\varepsilon)$ queries and total evolution time, notably avoiding any dependence on n. To do this, we reduce the problem of testing Hamiltonian sparsity to testing the sparsity of an associated Pauli channel. To be precise, given the time evolution channel $\mathcal{H}_t : \rho \to U(t)\rho U^\dagger(t)$, we define its *Pauli-twirled*

Table 2. Comparison of algorithms for learning Hamiltonians with $\|H\|_{\text{op}} \leq 1$. * It can be improved to $O(1/\varepsilon^{2+o(1)})$ total evolution time and $O(1/\varepsilon^{6+o(1)})$ queries by paying huge constant factors. † This algorithm works for Hamiltonians with $\sup_x |\lambda_x| \leq 1$, a weaker constraint than $\|H\|_{\text{op}} \leq 1$. ° This algorithm is the only one in the table that uses no quantum memory. We provide an algorithm with no quantum memory for k-local learning that performs as the one in the last row, but with an extra factor $\log n$.

Hamiltonians	Reference	t_{total}	Queries	Access model
Unstructured, ℓ_∞ error	[Car23]	n/ε^4	n/ε^4	Coherent queries
	Theorem 4.6	$1/\varepsilon^3$	$1/\varepsilon^4$	Incoherent queries
s-sparse, ℓ_∞ error	[Zha24]*	$1/\varepsilon^4$	$1/\varepsilon^8$	Coherent queries
	[HMG+25]†	s^2/ε	s^2/ε	Coherent queries
	Theorem 4.9	$1/\varepsilon^3$	$1/\varepsilon^4$	Incoherent queries
k-local, ℓ_2 error	[CW23]	n^k/ε^2	n^k/ε^2	Controlled and inverse queries
	[MFPT24]°	$(9n)^k/\varepsilon$	$(27n^3)^k/\varepsilon^2$	Coherent queries
	Theorem 4.7	$\exp(k^2)/\varepsilon^k$	$\exp(k^2)/\varepsilon^k$	Incoherent queries

channel via

$$\mathcal{H}_t^{\mathcal{T}}(\rho) = \mathbb{E}_x[\sigma_x \mathcal{H}_t(\sigma_x \rho \sigma_x)\sigma_x],$$

and we prove that $\mathcal{H}_t$ is sparse if and only if $\mathcal{H}_t^{\mathcal{T}}$ is sparse. Our result is then as follows. See Theorem 5.7 for a formal statement.

Result 1.9. *There is an algorithm with no quantum memory for Problem 1.2 for s-sparse Hamiltonians by making* $\text{poly}(s/\varepsilon)$ *queries to the evolution operator with* $\text{poly}(s/\varepsilon)$ *total evolution time.*

Lower bounds. One drawback of our learning and testing algorithms is the exponent of the sparsity parameter s, locality parameter k and the tolerance $(\varepsilon_2 - \varepsilon_1)$. Reducing to classical Boolean functions, we show lower bounds without memory that certify that the dependence on these parameters cannot be completely avoided, but an interesting and important future direction is to obtain the optimal results for these near-term relevant problems.[6]

Direct comparison to previous work. Comparing the plethora of Hamiltonian learning algorithms can be challenging due to the different assumptions on the structure of the Hamiltonians (local, sparse, geometrical structures, etc.), the different distances to measure the error (ℓ_∞ norm of the coefficients, ℓ_2 norm, etc.), the different complexity measures (queries, total evolution time, number of experiments, etc.), the different access models (coherent/non-coherent queries, with/without memory, etc.) and the different goals of the algorithm (minimizing the dependence on the dimension parameters like n, s, k, achieving the Heisenberg scaling $1/\varepsilon$, etc.). Thus, we only include a direct comparison in Table 2 with the works that explicitly consider the same structure and the same error metric as us. As a summary, one can say that for constant ε our results achieve better dependence on the parameters n, s, k than previous work, while also using the weaker model of incoherent queries, where one can perform only one query before measuring, as opposed to the coherent query model. We also want to remark that our result for learning unstructured Hamiltonians is time-efficient, while the only previous one is not [Car23].

[6] We remark that Bakshi et al. [BLMT24] used highly non-trivial ideas to get Heisenberg scaling for their learning algorithm, and potentially similar ideas could be useful here.

1.4. Discussion and open questions. Our work opens up several interesting directions which we state here and leave for future work.

1. **Dependence on parameters** $\varepsilon_1, \varepsilon_2$. Our main objective in this work was to obtain query complexities for testing and learning with good dependence on the structural parameters. It is natural to ask if we could improve the dependence on the error parameters and perhaps achieve Heisenberg limited scaling as has been shown to be possible in some particular cases for Hamiltonian learning [HTFS23,BLMT24].
2. **Robustness to SPAM noise.** It would be desirable to make the protocols introduced in this work robust to state-preparation and measurement (SPAM) noise. A potential approach is to adapt strategies in [FW20].
3. **Adaptivity.** For learning structured Hamiltonians, adaptive strategies [GFWC12, DPW+23] can improve query complexity by shedding constant factors over baseline learning algorithms, thereby improving performance in practice. Another direction is to then explore adaptive protocols for testing structured Hamiltonians and the performance gains they may bring.
4. **Testing and learning with limited quantum memory.** For estimating properties of quantum states, Chen et al. [CCHL22] showcased the utility of the resource of quantum memory or a k-qubit ancillary system ($k < n$). Large separations in query complexity when learning with memory (even for $k \ll n$) and without memory have been reported for learning Pauli channels [CZSJ22,CG23] and shadow tomography [CGY24]. We could thus imagine having access to only limited quantum memory during learning or testing structured Hamiltonians as well. However, it should be noted that given the separation between the query complexities (see Table 1) with access to n-qubit quantum memory and without any, only marginal gains in complexity are expected from having access to limited quantum memory.
5. **Testing and learning Hamiltonians from Gibbs states.** Another natural learning model is that of having access to copies of the Gibbs state of a quantum Hamiltonian at a certain inverse temperature. There has been a suite of work investigating learning local Hamiltonians from Gibbs states [AAKS21,BLMT23] but answering the question of testing structured Hamiltonians given access to copies of the Gibbs state remains wide open.

Note added. After sharing Theorem 4.1 with Bluhm et al., they independently improved the analysis of their testing algorithm and showed that it only requires $O(1/(\varepsilon_2-\varepsilon_1)^3\varepsilon_2)$ queries and $O(1/(\varepsilon_2 - \varepsilon_1)^{2.5}\varepsilon_2^{0.5})$ total evolution time, which is very similar to our Theorem 4.1 [BCO24b]. In addition, for a wide range of $k = O(n)$, their algorithm does not require the use of auxiliary qubits.

2. Preliminaries

2.1. Notation. In this section we collect a few well-known facts that we will repeatedly use in our proofs. The 1-qubit Pauli matrices matrices are defined as follows

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \text{ and } Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

It is well-known that the n-qubit Pauli matrices $\{I, X, Y, Z\}^{\otimes n}$ form an orthonormal basis for $\mathbb{C}^{2^n}$. In particular, for every $x = (a, b) \in \mathbb{F}_2^{2n}$, one can define the *Pauli operator*

$$\sigma_x = i^{a\cdot b}(X^{a_1}Z^{b_1} \otimes X^{a_2}Z^{b_2} \otimes \cdots \otimes X^{a_n}Z^{b_n}).$$

and these operators $\{\sigma_x\}_{x\in\mathbb{F}_2^{2n}}$ are orthonormal. Hence, every n-qubit operator H can be written down in its Pauli decomposition as

$$H = \sum_{x\in\{0,1\}^{2n}} \lambda_x \sigma_x,$$

where the real-valued coefficients λ_x are given by $\lambda_x = \frac{1}{2^n}\operatorname{Tr}(H\sigma_x)$. Parseval's identity states that the normalized Frobenius norm of H, denoted as $\|H\|_2$, is the ℓ_2-norm of its Pauli spectrum, i.e.,

$$\|H\|_2 = \sqrt{\frac{\operatorname{Tr}[H^\dagger H]}{2^n}} = \sqrt{\sum_{x\in\{0,1\}^{2n}} |\lambda_x|^2}.$$

We will repeatedly use that $\|H\|_2 \le \|H\|_{\mathrm{op}}$, which holds because $\|H\|_2^2$ is the average of the squares of the eigenvalues of H. We will also consider the ℓ_∞ norm of the Pauli coefficients of an operator, which is given by

$$\|H\|_{\ell_\infty} = \sup_x |h_x|.$$

Additionally, we will use $\|H\| := \max\{\|H\|_{\mathrm{op}}, 1\}$.

Throughout this work, we will also use the following correspondence between $\{0,1\}^{2n}$ and $\{0,1,2,3\}^n$: i.e., for $a, b \in \mathbb{F}_2^n$ and $(a,b) \in \{0,1\}^{2n}$, consider the string $\big((a_1,b_1), \ldots, (a_n,b_n)\big) \subseteq (\mathbb{F}_2^2)^n$ and one can write out (a_i, b_i) as an element in $\{0,1,2,3\}$.

Given $x \in \{0,1,2,3\}^n$, define $|x|$ as the number of indices $i \in [n]$ where $x_i \neq 0$, define

$$H_{>k} = \sum_{|x|>k} \lambda_x \sigma_x$$

and $H_{\le k}$ as $\sum_{|x|\le k} \lambda_x \sigma_x$. From the formulation of the 2-norm in terms of the Pauli coefficients, it follows that $\|H_{>k}\|_2 \le \|H\|_2$. We note that the distance of a Hamiltonian H from the space of k-local Hamiltonians is given by $\|H_{>k}\|_2$, as $H_{\le k}$ is the k-local Hamiltonian closest to H. The ℓ_2-distance of H to being s-sparse also has a nice expression. Assign labels from $[4^n]$ to $x \in \{0,1\}^{2n}$ in a way that and $|\lambda_{x_1}| \ge |\lambda_{x_2}| \cdots \ge |\lambda_{x_{4^n}}|$. Then, $\sum_{i\in[s]} \lambda_{x_i}\sigma_{x_i}$ is the closest s-sparse Hamiltonian to H, so the ℓ_2-distance of H to the space of s-sparse Hamiltonians is $\sqrt{\sum_{i>s} |\lambda_{x_i}|^2}$.

2.2. Necessary subroutines. Suppose U is a unitary and we write out its Pauli decomposition as $U = \sum_x \widehat{U}_x \sigma_x$, then by Parseval's identity $\sum |\widehat{U}_x|^2 = \operatorname{Tr}[U^\dagger U]/2^n = 1$, i.e., $\{|\widehat{U}_x|^2\}_x$ is a *probability distribution*. We will be using the fact below extensively.

Fact 2.1. *Given access to a unitary U, one can sample from the distribution $\{|\widehat{U}_x|^2\}_x$.*

Proof. The proof simply follows by applying $U \otimes \mathrm{Id}_{2^n}$ to n EPR pairs (i.e., preparing the Choi-Jamiolkowski state of U) and measuring in the Bell basis, because

$$U \otimes \mathrm{Id}_{2^n} |\mathrm{EPR}_n\rangle = \sum_{x\in\{0,1\}^{2n}} \widehat{U}_x \bigotimes_{i\in[n]} (\sigma_{x_i} \otimes \mathrm{Id}_2 |\mathrm{EPR}\rangle),$$

and the Bell states can be written as $\sigma_x \otimes \mathrm{Id}_2 |\mathrm{EPR}\rangle$ for $x \in \{0,1,2,3\}$. □

We will also use that given a Hamiltonian H, the Taylor expansion of the exponential allows us to approximate the time evolution operator as

$$U(t) = e^{-itH} = \mathrm{Id}_{2^n} - itH + ct^2 R_1(t)\|H\|_{\mathrm{op}}^2 \tag{1}$$

for $t \leq 1/(2\|H\|_{\mathrm{op}})$, where the first order remainder $R_1(t)$ is bounded $\|R_1(t)\|_{\mathrm{op}} \leq 1$ and $c > 1$ is a constant. For more background on the Taylor expansion of functions of matrices, see [Hig08, Section 4.3].

Fact 2.2 ([Can20]). *Let $p : \mathcal{X} \to [0, 1]$ be a distribution for a finite set $\mathcal{X}$. Then, with prob. $\geq 1 - \delta$,*

$$O\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$$

samples from p, the empirical distribution $\tilde{p}$ satisfies $\max_{x\in\mathcal{X}} |p(x) - \tilde{p}(x)| \leq \varepsilon$

Theorem 2.3 (Clifford shadows [HKP20]). *Let ρ be an n-qubit state and let $\{O_i\}_{i\in[M]}$ be n-qubit traceless observables. Assume that $\sup_i \mathrm{Tr}[O_i^2] = O(1)$. Then, Algorithm 1 obtains estimates $\widetilde{O}_{i,\rho}$ such that, with probability $1 - \delta$, satisfy*

$$|\mathrm{Tr}[O_i\rho] - \widetilde{O}_{i,\rho}| \leq \varepsilon$$

for every $i \in [M]$. The algorithm uses $O\left(\frac{\log(M/\delta)}{\varepsilon^2}\right)$ copies of ρ.

Algorithm 1 Clifford shadows

Input: Copies of a quantum state ρ, target set of observables $\{O_i\}_{i\in[M]}$, error parameter $\varepsilon \in (0, 1)$, and failure parameter $\delta \in (0, 1)$

Set $T = O(\log(M/\delta)/\varepsilon^2)$ and $J = O(\log(M/\delta))$
for $j \in [J]$ **do**
  for $k \in [T/J]$ **do**
    Apply a uniformly random Clifford gate C to a copy of ρ
    Measure in the computational basis. Set $|b_{j,k}\rangle$ be the outcome
    for $i \in [M]$ **do**
      Set $\widetilde{O}_{i,j,k} = (2^n + 1)\langle b_{j,k}|C^{-1} O_i C|b_{j,k}\rangle$
for $i \in [M]$ **do**
  Set $\widetilde{O}_{i,j} = \mathrm{Mean}((\widetilde{O}_{i,j,k})_k)$, ▷ *Mean* is the arithmetic mean of $(\widetilde{O}_{i,j,1}, \ldots, \widetilde{O}_{i,j,T/J})$
for $i \in [M]$ **do**
  Set $\widetilde{O}_i := \mathrm{Median}((O_{i,j})_j)$

Output: $(\widetilde{O}_i)_{i\in[M]}$

2.3. Concentration inequalities. We state a few concentration inequalities that we use often.

Lemma 2.4 (Hoeffding bound). *Let $X_1, \ldots, X_m$ be independent-random variables that satisfy $-a_i \leq |X_i| \leq a_i$ for some $a_i > 0$. Then, for any $\tau > 0$, we have*

$$\Pr\left[\left|\sum_{i\in[m]} X_i - \sum_{i\in[m]} \mathbb{E}[X_i]\right| > \tau\right] \leq 2\exp\left(-\frac{\tau^2}{2(a_1^2 + \cdots + a_m^2)}\right).$$

Lemma 2.5 (Bernstein inequality). *Let $X_1, \ldots, X_m$ be independent-random variables with $|X_i| \leq M$ for some $M > 0$. Then,*

$$\Pr\Big[\Big|\sum_{i\in[m]} X_i - \sum_{i\in[m]} \mathbb{E}[X_i]\Big| > \tau\Big] \leq 2\exp\left(-\frac{\tau^2/2}{\sum_{i\in[m]} \mathrm{Var}[X_i] + \tau M/3}\right).$$

2.4. Symplectic Fourier analysis. Consider $x, y \in \mathbb{F}_2^{2n}$ with $x = (x_1, x_2)$ and $y = (y_1, y_2)$. Define the symplectic inner product as

$$[x, y] = \langle x_1, y_2\rangle + \langle x_2, y_1\rangle \mod 2. \tag{2}$$

Under this notation, observe that

$$\sigma_x\sigma_y = (-1)^{[x,y]}\sigma_y\sigma_x. \tag{3}$$

The symplectic Fourier decomposition for functions $f : \mathbb{F}_2^{2n} \to \mathbb{R}$ is defined as

$$f(x) = \sum_{a\in\mathbb{F}_2^{2n}} (-1)^{[a,x]}\, \breve{f}(a),$$

where the symplectic Fourier coefficients are defined as

$$\breve{f}(a) = \frac{1}{4^n}\sum_{x\in\mathbb{F}_2^{2n}} (-1)^{[a,x]} f(x).$$

2.5. Subspaces, stabilizer groups and stabilizer states. Let $V \subseteq \{0,1\}^{2n}$ be a set of strings corresponding to Paulis. We will denote $\mathsf{C}(V)$ as the commutant of the set V (also called centralizer of V when V is a group), defined as $\mathsf{C}(V) = \{c : [c, h] = 0\ \forall h \in V\}$. We define $\mathsf{A}(V)$ to be the quotient space $\mathbb{F}_2^{2n}/\mathsf{C}(V)$.

Fact 2.6. *Suppose V is a subspace of $\mathbb{F}_2^{2n}$. Then,*

$$\frac{1}{|V|}\sum_{x\in V}(-1)^{[a,x]} = [a \in \mathsf{C}(V)],$$

where $[\cdot]$ is the indicator of the event in the parenthesis.

We will denote a stabilizer state corresponding to a stabilizer subgroup G of dimension $k \leq n$ as

$$\rho_G = \frac{1}{2^n}\sum_{g\in G}\sigma_g, \tag{4}$$

which will be a pure stabilizer state when dimension of G is $k = n$ i.e., G is an n-qubit stabilizer group. When $k < n$, then this will be a mixed state. Another useful state will be the state obtained by action of a Pauli σ_x on ρ_G which we will denote by $\rho_{G,x}$ and is given by

$$\rho_{G,x} = \sigma_x\rho_G\sigma_x = \frac{1}{2^n}\sum_{g\in G}(-1)^{[x,g]}\sigma_g. \tag{5}$$

Note that if $x \in \mathsf{C}(G)$, then $\rho_{G,x}$ equals ρ_G.

2.6. Pauli Channels. An n-qubit channel $\mathcal{E}$ is called a *Pauli channel* if it can be written as

$$\mathcal{E}(\rho) = \sum_{x\in\{0,1\}^{2n}} p(x)\sigma_x\rho\sigma_x, \tag{6}$$

where $p(x)$ are referred to as the *error rates* corresponding to $x \in \{0,1\}^{2n}$. The eigenvalues of a Pauli channel, also called *Pauli fidelities*, are easy to compute as follows

$$\lambda(y) = \frac{1}{2^n}\operatorname{Tr}\left(\sigma_y \cdot \mathcal{E}(\sigma_y)\right), \tag{7}$$

where we have denoted the Pauli fidelity corresponding to $y \in \{0,1\}^{2n}$ as $\lambda(y)$. Of relevance to us is that the Pauli fidelities and error rates are related via the symplectic Fourier transform as $\breve{\lambda}(\alpha) = p(\alpha)$ for all $\alpha \in \{0,1\}^{2n}$. We define $\mathrm{Spec}(\mathcal{E}) = \{x \in \{0,1\}^{2n} : p(x) \neq 0\}$.

We say that a Pauli channel is s-sparse if the corresponding set of Pauli coefficients (or error rates) $\{p(x)\}_{x\in\mathbb{F}_2^{2n}}$ contains at most s non-zero values. We define the *energy* of the top s Pauli coefficients of a given Pauli channel $\mathcal{E}$ as $\mathsf{Energy}(\mathcal{E}; s)$, expressed as

$$\mathsf{Energy}(\mathcal{E}; s) := \max_{\substack{T\subseteq\mathbb{F}_2^{2n}:\\ |T|=s}} \left\{\sum_{x\in T} p(x)\right\}. \tag{8}$$

We denote the distance between two Pauli channels $\mathcal{E}_1$ (with error rates $\{p_1(x)\}$) and $\mathcal{E}_2$ (with error rates $\{p_2(x)\}$) as the total variation distance between the error rates, namely

$$\mathrm{dist}(\mathcal{E}_1, \mathcal{E}_2) = \frac{1}{2}\|p_1 - p_2\|_1.$$

This distance is equivalent to the diamond distance up a factor 2 [MGE12, Section A]. We now give a simple formula for the distance of Pauli channels to the set of sparse Pauli channels.

Claim 2.7. *Let $\mathcal{E}$ be an n-qubit Pauli channel. Then, distance $d(\mathcal{E}, s\text{-sparse})$ of $\mathcal{E}$ to the set of s-sparse Pauli channels satisfies*

(a) $d(\mathcal{E}, s\text{-sparse}) \leq \varepsilon_1 \implies \mathsf{Energy}(\mathcal{E}; s) \geq 1 - 2\varepsilon_1$,
(b) $d(\mathcal{E}, s\text{-sparse}) \geq \varepsilon_2 \implies \mathsf{Energy}(\mathcal{E}; s) \leq 1 - \varepsilon_2$.

Proof. We begin with (a). If $d(\mathcal{E}, s\text{-sparse}) \leq \varepsilon_1$, then there is a s-sparse probability distribution $\{q(x)\}_x$ such that

$$\sum_{x\in\{0,1\}^{2n}} |p(x) - q(x)| \leq 2\varepsilon_1.$$

In particular, if S is the support of $q(x)$, then $\sum_{x\notin S}|p(x)| \leq 2\varepsilon_1$, so $\sum_{x\in S}|p(x)| \geq 1 - 2\varepsilon_1$, as desired. Now, we prove (b). Assume $d(\mathcal{E}, s\text{-sparse}) \geq \varepsilon_2$. Let $S \subseteq \{0,1\}^{2n}$

be the set of the s largest Pauli coefficients of $\mathcal{E}$. Then, $\mathcal{E}'(\cdot) = \sum_{x\in S} \frac{p(x)}{\mathsf{Energy}(\mathcal{E},s)}\sigma_x \cdot \sigma_x$ is a s-sparse Pauli channel. As such, it satisfies that

$$\begin{aligned} 2\varepsilon_2 &\leq \sum_{x\in S}\left|p(x) - \frac{p(x)}{\mathsf{Energy}(\mathcal{E},s)}\right| + \sum_{x\notin S} p(x) \\ &= \frac{1-\mathsf{Energy}(\mathcal{E},s)}{\mathsf{Energy}(\mathcal{E},s)}\sum_{x\in S} p(x) + \left(1 - \sum_{x\in S} p(x)\right) \\ &= 2(1-\mathsf{Energy}(\mathcal{E},s)), \end{aligned}$$

where in the first line we have used that $\mathsf{Energy}(\mathcal{E},s) \leq 1$ and that $\sum_{x\in\{0,1\}^{2n}} p(x) = 1$. Now, (b) follows by dividing by 2 in both sides. □

3. Technical Results

In this section, we will first prove our main structural theorems for Hamiltonians and provide subroutines which will be used later for testing and learning these structured Hamiltonians.

3.1. Structural lemma for local Hamiltonians.

First, we prove a lemma regarding the discrepancy on the weights of non-local terms of the short-time evolution operator for close-to-local and far-from-local Hamiltonians.

Lemma 3.1. *Let $0 \leq \varepsilon_1 < \varepsilon_2$. Let H be an n-qubit Hamiltonian and $\alpha = (\varepsilon_2 - \varepsilon_1)/(3c\|H\|^2)$, where c is the constant appearing in Eq. (1). If H is ε_1-close to k-local, then*

$$\|U(\alpha)_{>k}\|_2 \leq (\varepsilon_2 - \varepsilon_1)\frac{2\varepsilon_1 + \varepsilon_2}{9c\|H\|^2},$$

and if H is ε_2-far from being k-local, then

$$\|U(\alpha)_{>k}\|_2 \geq (\varepsilon_2 - \varepsilon_1)\frac{\varepsilon_1 + 2\varepsilon_2}{9c\|H\|^2}.$$

In particular, if H is ε_2-far from local, then the quantity $\|U(\alpha)_{>k}\|_2^2$ is $\left((\varepsilon_2-\varepsilon_1)^2/(9c\|H\|^2)\right)^2$ larger than if H is ε_1-close to local.

Proof. Recall that, as $\alpha \leq 1/(2\|H\|_{\mathrm{op}})$, then $U(\alpha) = \mathrm{Id}_{2^n} - i\alpha H + c\alpha^2 R(\alpha)\|H\|_{\mathrm{op}}^2$ by Eq (1) where $\|R\|_{\mathrm{op}} \leq 1$. For simplicity, we set $U = U(\alpha)$ and $R = R_1(\alpha)$. First, assume that H is ε_1-close to k-local, then by definition we have that $\|H_{>k}\|_2 \leq \varepsilon_1$. Then

$$\begin{aligned} \|U_{>k}\|_2 &\leq \alpha\|H_{>k}\|_2 + c\alpha^2\|R_{>k}\|_2\|H\|_{\mathrm{op}}^2 \leq \frac{\varepsilon_2-\varepsilon_1}{3c\|H\|^2}\varepsilon_1 + c\left(\frac{\varepsilon_2-\varepsilon_1}{3c\|H\|}\right)^2 \\ &= (\varepsilon_2-\varepsilon_1)\frac{2\varepsilon_1+\varepsilon_2}{9c\|H\|^2}, \end{aligned}$$

where in the first inequality we have used the triangle inequality, and in the second that H is ε_1-close to k-local and that $\|R_{>k}\|_2 \leq \|R\|_2 \leq \|R\|_{\text{op}} \leq 1$. Now, assume that H is ε_2-far from being k-local (i.e., $\|H_{>k}\|_2 \geq \varepsilon_2$). Then

$$\begin{aligned}\|U_{>k}\|_2 &\geq \alpha\|H_{>k}\|_2 - c\alpha^2\|R_{>k}\|_2\|H\|_{\text{op}}^2 \geq \frac{\varepsilon_2-\varepsilon_1}{3c\|H\|^2}\varepsilon_2 - c\left(\frac{\varepsilon_2-\varepsilon_1}{3c\|H\|}\right)^2 \\ &\geq (\varepsilon_2-\varepsilon_1)\frac{\varepsilon_1+2\varepsilon_2}{9c\|H\|^2},\end{aligned}$$

where in first inequality we have used triangle inequality on $i\alpha H = c\alpha^2\|H\|^2 R(\alpha) - U(\alpha)$ to conclude $\alpha\|H_{>k}\|_2 \leq \|U_{>k}\|_2 + c\alpha^2\|R_{>k}\|_2$, and in the second the fact that H is ε_2-far from k-local. □

3.2. Structural lemma for sparse Hamiltonians. Similar to local Hamiltonians, we show a discrepancy in the sum of the top Pauli coefficients of the short-time evolution operator for close-to-sparse and far-from-sparse Hamiltonians. To formally state this result we need to introduce the concept of *top energy*. Let $U(t)$ the time evolution operator at time t and let $\{\widehat{U}(t)\}_x$ be its Pauli coefficients. We assign labels from $\{x_0, \ldots, x_{4^n-1}\}$ to $x \in \{0,1\}^{2n}$ in a way that $\widehat{U}_{x_0} = \widehat{U}_{0^n}$ and $|\widehat{U}_{x_1}| \geq |\widehat{U}_{x_2}| \geq \cdots \geq |\widehat{U}_{x_{4^n-1}}|$. Now, we define the top energy at time t as

$$\mathsf{TopEnergy}(t;s) := |\widehat{U}_{x_0}(t)|^2 + \sum_{i\in[s]} |\widehat{U}_{x_i}(t)|^2,$$

Lemma 3.2. *Let H be a n-qubit Hamiltonian with $\text{Tr}[H] = 0$. Let $t \in (0, 1/2\|H\|_{\text{op}})$. On the one hand, if H is ε_1-close to s-sparse, then*

$$\mathsf{TopEnergy}(t;s) \geq 1 - \varepsilon_1^2 t^2 - O(t^3\|H\|_{\text{op}}^3 s).$$

On the other hand, if H is ε_2-far from s-sparse, then

$$\mathsf{TopEnergy}(t;s) \leq 1 - \varepsilon_2^2 t^2 + O(t^3\|H\|_{\text{op}}^3 s).$$

Proof. For this proof we need to consider the 2nd order Taylor expansion of $U(t)$,

$$U(t) = \text{Id} - itH - t^2H^2/2 + O(t^3\|H\|_{\text{op}}^3)R_2,$$

for $t \leq 1/(2\|H\|_{\text{op}})$ where R_2 is the remainder of the series of order 2 that satisfies $\|R_2\|_{\text{op}} \leq 1$. Since $\text{Tr}[H] = 0$ (so $\lambda_{0^n} = 0$), we have

$$\widehat{U}_0(t) = 1 - \frac{t^2}{2}\cdot\sum_{x\in\{0,1\}^{2n}}\lambda_x^2 + O(t^3\|H\|_{\text{op}}^3),$$

so, using that $|a^2 - b^2| = |a-b||a+b|$, we have that

$$\left||\widehat{U}_0(t)|^2 - \left(1 - t^2\sum_{x\in\{0,1\}^{2n}}\lambda_x^2\right)\right| = O(t^3\|H\|_{\text{op}}^3). \tag{9}$$

To control $|U_x(t)|$ for $x \neq 0^{2n}$, we use the first order Taylor expansion of $U(t) = \mathrm{Id}_{2^n} - itH + ct^2 R_1(t)\|H\|_{\mathrm{op}}^2$, for $t \leq 1/2\|H\|_{\mathrm{op}}$, and get

$$\begin{aligned}\left||\widehat{U}_x(t)| - |t\lambda_x|\right| &\leq |\widehat{U}_x(t) - (-it\lambda_x)| \leq \|U(t) - (-itH)\|_2 \leq O(t^2\|H\|_{\mathrm{op}}^2)\|R_1\|_2 \\ &\leq O(t^2\|H\|_{\mathrm{op}}^2), \end{aligned} \tag{10}$$

where we again used that $\|R_1\|_2 \leq 1$. From this, it follows that

$$\begin{aligned}\left||\widehat{U}_x(t)|^2 - t^2\lambda_x^2\right| &= \left|\left(|\widehat{U}_x(t)| - |t\lambda_x|\right) \cdot \left(|\widehat{U}_x(t)| + |t\lambda_x|\right)\right| \\ &= O(t^2\|H\|_{\mathrm{op}}^2)(|U_x| + |t\lambda_x|) \\ &= O(t^2\|H\|_{\mathrm{op}}^2)(2|t\lambda_x| + O(t^2\|H\|_{\mathrm{op}}^2)) \\ &= O(t^3\|H\|_{\mathrm{op}}^3), \end{aligned}$$

where the second and third equality both used Eq. (10); and in the last line used $|\lambda_x| \leq \|H\|_{\mathrm{op}}$. In particular, the above implies that

$$|\widehat{U}_x(t)|^2 \geq t^2|\lambda_x|^2 - O(t^3\|H\|_{\mathrm{op}}^3) \tag{11}$$

Now we will define a quantity similar to the top energy, but now we will define the top coefficients as the top coefficients of H. To be precise, we assign labels to $\{y_0, \ldots, y_{4^n-1}\}$ to the elements of $\{0,1\}^{2n}$ in a way such that $y_0 = 0^{2n}$ and $|\lambda_{y_1}| \geq \cdots \geq |\lambda_{y_{4^n-1}}|$. We now define

$$\mathsf{TopEnergy}_H(t;s) := \left(1 - t^2 \sum_{x\in\{0,1\}^{2n}} \lambda_x^2\right) + \sum_{i\in[s]} (t\lambda_{y_i})^2.$$

If the top s Pauli coefficients of H coincided with the ones of $U(t)$ and there was no error in the Taylor expansion, then $\mathsf{TopEnergy}_H(t;s)(t) = \mathsf{TopEnergy}(t;s)$. However, this may not be true in general. Nevertheless, we show that both quantities are close to each other. To this end,

$$\begin{aligned}\mathsf{TopEnergy}(t;s) &= |\widehat{U}_{x_0}(t)|^2 + \sum_{i\in[s]} |\widehat{U}_{x_i}(t)|^2 \\ &\geq |\widehat{U}_{y_0}(t)|^2 + \sum_{i\in[s]} |\widehat{U}_{y_i}(t)|^2 \\ &\geq \left(1 - t^2 \sum_{x\in\{0,1\}^{2n}} \lambda_x^2\right) + \sum_{i\in[s]} (t\lambda_{y_i})^2 - (s+1)O(t^3\|H\|_{\mathrm{op}}^3) \\ &= \mathsf{TopEnergy}_H(t;s) - (s+1)O(t^3\|H\|_{\mathrm{op}}^3), \end{aligned}$$

where in the first inequality we used that $x_1, \ldots, x_s$ correspond to the s largest coefficients of $U(t)$, so $\sum_{i\in[s]} |\widehat{U}_{x_i}(t)|^2$ is larger than the sum of the squares of any other s coefficients of U; in the second inequality we used Eqs. (9) and (11). Similarly, one can check that $\mathsf{TopEnergy}_H(t;s) \geq \mathsf{TopEnergy}(t;s) - (s+1)O(t^3\|H\|_{\mathrm{op}}^3)$, so

$$|\mathsf{TopEnergy}_H(t;s) - \mathsf{TopEnergy}(t;s)| \leq O(st^3\|H\|_{\mathrm{op}}^3).$$

Now, the claimed result follows by noticing that

$$\mathsf{TopEnergy}_H(t;s) = 1 - t^2 \sum_{i>s} |\lambda_{y_i}|^2,$$

and that $\sum_{i>s} |\lambda_{y_i}|^2$ is the square of the ℓ_2-distance of H to the space of s-sparse Hamiltonians, because $\sum_{i\in[s]} \lambda_{y_i}\sigma_{y_i}$ is the s-sparse Hamiltonian closest to H. □

3.3. Subroutines for learning without memory. Motivated by the difficulty of accessing quantum memory in the NISQ era [Pre18], we propose two subroutines that serve the purpose of substituting Pauli sampling (i.e., sampling from $\{|\widehat{U}_x|^2\}_x$ by creating the Choi state corresponding to U as in Fact 2.1) and the SWAP test in our learning algorithms, by protocols which do not require memory.

3.3.1. Estimating Pauli distribution Our first lemma constructs an algorithm to estimate the Pauli distribution determined by a n-qubit unitary in ℓ_∞-error with just $O(n)$ queries to the unitary. These queries are performed on a random stabilizer state and the measurement are also random stabilizer measurements, as in other quantum protocols used for testing and learning [FW20, Yu21, FOS23, BCO24b]. Here we propose a novel classical post-processing that allows us to emulate Pauli sampling in our context and others. In particular, we propose an algorithm to tolerantly test if a unitary is a k-junta, making progress in a question by Chen, Nadimpalli and Yuen [CNY23, Section 1.3], and show that can be implemented without quantum memory.

Lemma 3.3. *Let U be a n-qubit unitary, let $\mathcal{S} \subseteq \{0,1\}^{2n}$. There is an memory-less algorithm that makes $O\big(\log(|\mathcal{S}|/\delta)/\varepsilon^2\big)$ queries to the unitary and provides estimates $|\alpha_x|^2$ such that*

$$\big||\widehat{U}_x|^2 - |\alpha_x|^2\big| \leq \varepsilon$$

for every $x \in \mathcal{S}$. Furthermore, the algorithm only makes queries on stabilizer states and only performs Clifford measurements.

To this end, we will look at its Pauli expansion from the $\{0,1\}^{2n}$ point of view. Namely, we will consider the expansion

$$U = \sum_{x\in\{0,1\}^{2n}} \widehat{U}_x \sigma_x,$$

where $\sigma_{(a,b)} = i^{a\cdot b} X^a Z^b$. Before presenting our algorithm, we introduce a few facts, whose proofs can be found in [BBRV02]. Let $N = 2^n$. There exists subspaces $G_1, \ldots, G_{N+1} \subseteq \{0,1\}^{2n}$ such that

- $G_i = \{x \in \{0,1\}^{2n} : [x,y] = 0,\ \forall y \in G_i\}, \forall i \in [N+1]$,
- $G_i \cap G_j = \{\sigma_{0^{2n}}\}$ if $i \neq j$,
- $|G_i| = N$.

Let $i \in [N]$. Then, $|\{0,1\}^{2n}/G_i| = N$. Let $r_j^i \in \{0,1\}^{2n}$ for $j \in [N]$ be representatives of the different equivalence classes of $\{0,1\}^{2n}/G_i$. Then, for every $i, j \in [N]$ the following matrix determines a pure state

$$|\phi_{i,j}\rangle\langle\phi_{i,j}| = \frac{1}{N}\sum_{x\in G_i} (-1)^{[r_j^i, x]}\sigma_x. \tag{12}$$

An important property is that $\mathcal{B}_i = \{|\phi_{i,j}\rangle\}_{j\in[N]}$ is an orthonormal basis of $\mathbb{C}^{2^n}$ for every $i \in [N+1]$. Also, $\mathcal{B}_i$ are mutually unbiased bases and form a 2-design, meaning that

$$\frac{1}{N(N+1)} \sum_{i\in[N+1]} \sum_{j\in[N]} |\phi_{i,j}\rangle\langle\phi_{i,j}| \otimes |\phi_{i,j}\rangle\langle\phi_{i,j}| = \frac{\mathbb{I}+F}{N(N+1)}, \tag{13}$$

where F is the swap operator, i.e., $F = \sum |xy\rangle\langle yx|$ [BCO24b, Proposition 2.3]. We make an extra remark: given $i \in [N]$, $j \in [N+1]$ and $x \in \{0,1\}^{2n}$, there exists a unique $\ell(i,j,x)$ such that

$$|\langle\phi_{i,\ell(i,j,x)}|\sigma_x|\phi_{i,j}\rangle| = 1. \tag{14}$$

Indeed,

$$\sigma_x|\phi_{i,j}\rangle\langle\phi_{i,j}|\sigma_x = \sum_{y\in G_i} (-1)^{[r^i_j,y]}\sigma_x\sigma_y\sigma_x = \sum_{y\in G_i} (-1)^{[r^i_j+x,y]}\sigma_y,$$

so $\sigma_x|\phi_{i,j}\rangle\langle\phi_{i,j}|\sigma_x = |\phi_{i,\ell(i,j,x)}\rangle\langle\phi_{i,\ell(i,j,x)}|$ for $\ell(i,j,x)$ such that $r^i_{\ell(i,j,x)} \sim_{G_i} r^i_j + x$. Now, we are ready to introduce Algorithm 2 and prove Lemma 3.3.

Algorithm 2 Memory-less Pauli sampling

Input: Query access to a unitary U, error parameter $\varepsilon \in (0,1)$, a set $\mathcal{S} \subseteq \{0,1\}^{2n}$, failure parameter $\delta \in (0,1)$

Set $T = O(\log(|\mathcal{S}|/\delta)/\varepsilon^2)$
Initialize $|\alpha_x|^2 = |\beta_x|^2 = 0$ for $x \in \mathcal{S}$
for $t = 1, ..., T$ **do**
  Sample $i \in [N+1]$ and $j \in [N]$ uniformly at random
  Prepare $U|\phi_{i,j}\rangle$
  Measure in the basis $\mathcal{B}_i$ and let l be the outcome
  for $x \in \mathcal{S}$ **do**
    if $l = l(i,j,x)$ **then**
      $|\beta_x|^2 \leftarrow |\beta_x|^2 + 1/T$
for $x \in \mathcal{S}$ **do** $|\alpha_x|^2 = (N+1)/N|\beta_x|^2 - 1/N$

Output: $(|\alpha_x|^2)_{x\in\mathcal{S}}$

Proof of Lemma 3.3. One iteration of Algorithm 2 consists of the following:

- Pick uniformly at random $i \in [N+1]$ and $j \in [N]$:
- Prepare $U|\phi_{i,j}\rangle$:
- Measure the state in the basis according $\mathcal{B}_i$. Suppose we obtain outcome ℓ.

For every $x \in \{0,1\}^{2n}$, we define the following random variable that takes different values depending on the outcome of the random iteration we just have described:

$$M_x = \begin{cases} 0 & \text{if } \ell \neq \ell(i,j,x), \\ 1 & \text{if } \ell = \ell(i,j,x). \end{cases} \tag{15}$$

Note that given that M_x takes values within a bounded interval, by the Hoeffding bound and a union bound, we can estimate the expectation of M_x for every $x \in \mathcal{S}$ within error ε with probability $\geq 1-\delta$ from $O(\log(|\mathcal{S}|/\delta)/\varepsilon^2)$ repetitions of the iteration above,

which requires 1 query and no quantum memory. Thus, it only remains to show that these expectations are closely related to $|\widehat{U}_x|^2$. To this end, observe that

$$\begin{aligned}
\mathbb{E}[M_x] &= \mathbb{P}[\ell = \ell(i,j,x)] \\
&= \frac{1}{N(N+1)} \sum_{i,j} |\langle \phi_{i,\ell(i,j,x)}|U|\phi_{i,j}\rangle|^2 \\
&= \frac{1}{N(N+1)} \sum_{i,j} |\langle \phi_{i,j}|\sigma_x U|\phi_{i,j}\rangle|^2 \\
&= \frac{1}{N(N+1)} \operatorname{Tr}\left((\sigma_x U)^\dagger \otimes \sigma_x U \cdot \sum_{i,j} |\phi_{i,j}\rangle\langle\phi_{i,j}|^{\otimes 2}\right) \\
&= \frac{1}{N(N+1)} \operatorname{Tr}\left((\sigma_x U)^\dagger \otimes \sigma_x U \cdot (I+F)\right) \\
&= \frac{1}{N(N+1)} (\operatorname{Tr}[U^\dagger \sigma_x \sigma_x U] + |\operatorname{Tr}[\sigma_x U]|^2) \\
&= \frac{1}{N+1} + \frac{N}{N+1} |\widehat{U}_x|^2,
\end{aligned}$$

where in the third line we used the definition of $\ell(i,j,x)$ (see Eq. (14)), in the fourth line we have used Eq. (13). The equation above implies that,

$$|\widehat{U}_x|^2 = \frac{N+1}{N}\mathbb{E}[M_x] - \frac{1}{N},$$

so $((N/N+1)\varepsilon)$-estimates of M_x yield ε-estimates of $|\widehat{U}_x|^2$.

Bonus application: Testing k-junta quantum unitaries. Now, we use Pauli sampling to show that $O(\exp(k)\operatorname{poly}(1/(\varepsilon_2-\varepsilon_1)))$ queries are sufficient to test whether an n-qubit unitary is ε_2-far or ε_1-close to k-junta. In addition, using Lemma 3.3 shows that $O(n\exp(k)\operatorname{poly}(1/(\varepsilon_2-\varepsilon_1)))$ queries are sufficient for the same task in the absence of memory. To the best of our knowledge, this is the first tolerant k-junta tester for unitaries. Despite $\tilde{O}(\sqrt{k})$ queries being sufficient for non-tolerant testing [CNY23], the worse k-dependence of our tolerant tester should not come as a surprise, as classical tolerant testing of k-junta Boolean functions requires $\Omega(2^{\sqrt{k}})$ samples, but the non-tolerant tester requires only $O(k)$ samples [Bla09,PRW22]. We will make use of the following lemma by Wang [Wan11].

Lemma 3.4 ([Wan11]). *Let U be an n-qubit unitary. If U is ε_1-close to k-junta in 2-norm, then there exists $K \subseteq [n]$ of size k such that*

$$\sum_{\mathbf{supp}(x)\subseteq K} |\widehat{U}_x|^2 \geq 1-\varepsilon_1^2.$$

On the other hand, if U is ε_2-far from k-local in 2-norm, then for every $K \subseteq [n]$ of size k

$$\sum_{\mathbf{supp}(x)\subseteq K} |\widehat{U}_x|^2 \leq 1-\frac{\varepsilon_2^2}{4}.$$

Proposition 3.5. *Let U be an n-qubit unitary and $0 < 2\varepsilon_1 < \varepsilon_2$. One can test whether U is ε_2-far or ε_1-close to k-junta in 2-norm by making $O(16^k/(\varepsilon_2 - \varepsilon_1)^4)$ queries to U with n qubits of memory. In absence of memory, $O(16^k n/(\varepsilon_2 - \varepsilon_1)^4)$ queries are sufficient.*

Proof. Assume for the moment that we have a distribution $(\alpha_x)_{x\in\{0,1,2,3\}^n}$ such that

$$|\alpha_x - |\widehat{U}_x|^2| \leq \frac{\varepsilon_2^2/4 - \varepsilon_1^2}{2 \cdot 4^k} \tag{16}$$

for every $x \in \{0, 1, 2, 3\}^n$. Then, for every $K \subseteq [n]$ of size k we can approximate $\sum_{\mathrm{supp}(x)\subseteq K} |\widehat{U}_x|^2$ via $\sum_{\mathrm{supp}(x)\subseteq K} \alpha_x$ up to error $(\varepsilon_2^2/4 - \varepsilon_1^2)/2$. Thanks to Lemma 3.4, this is enough for testing if U is ε_2-close to or ε_1-far from local.

If we have access to n-qubits of quantum memory, then we can sample $O(16^k/(\varepsilon_2^2 - \varepsilon_1^2)^2)$ times from $(|\widehat{U}_x|^2)_x$ with $O(16^k/(\varepsilon_2^2-\varepsilon_1^2)^2)$ queries via Fact 2.1, and the empirical distribution will satisfy Eq. (16) thanks to Fact 2.2. If we have no access to quantum memory, then $O(16^k n/(\varepsilon_2^2 - \varepsilon_1^2)^2)$ queries are sufficient thanks to Lemma 3.3. □

3.3.2. Estimating Pauli coefficients Below we give a protocol that allows one to estimate the Pauli coefficients of the unknown unitary without quantum memory. It is based on the fact that any non-identity Pauli operator can be written as the product of two other anti-commuting Pauli operators, a fact which has been previously used in [Car23, Lemma 6.1].

Lemma 3.6. *Let $x \in \{0, 1\}^{2n}$ and let H be an n-qubit traceless Hamiltonian. There is a memory-less algorithm (Algorithm 3) that by making $O(\|H\|^4/\varepsilon^4)$ queries to $U(\varepsilon/\|H\|^2)$ on a product state and making Pauli measurements can output an estimate $\tilde{\lambda}_x$ such that $|\lambda_x - \tilde{\lambda}_x| \leq \varepsilon$.*

Algorithm 3 Memory-less Pauli coefficient estimation

Input: Query access to the time evolution of $U(t) = e^{-itH}$, error parameter $\varepsilon \in (0, 1)$, $x \in \{0, 1\}^{2n}$, failure parameter $\delta \in (0, 1)$

1: Set $T = O((\|H\|^4/\varepsilon^4)\log(1/\delta))$
2: Pick $x', x'' \in \{0, 1\}^{2n}$, such that $\sigma_{x'}\sigma_{x''} = a\sigma_x$ for some $a \in \{\pm i\}$, and $\sigma_{x'}\sigma_{x''} = -\sigma_{x''}\sigma_{x'}$. Set $\alpha = 0$
3: **for** $j = 1, ..., T$ **do**
4: Prepare $\rho = (\mathrm{Id} - \sigma_{x'})/2^n$
5: Apply $U(\Theta(\varepsilon/\|H\|^2))$ to ρ
6: Measure in a eigenbasis of $\sigma_{x''}$. Let α_j be the ± 1 eigenvalue of the measured eigenvector of $\sigma_{x''}$
7: $\alpha \leftarrow \alpha + \alpha_j/T$

Output: $\alpha/(2i\varepsilon a)$

Proof. Let $x', x'' \in \{0,1\}^{2n}$ and $a \in \{-i, i\}$ such that $\sigma_{x'}\sigma_{x''} = a\sigma_x$ and $\sigma_{x'}\sigma_{x''} = -\sigma_{x''}\sigma_{x'}$. Then,

$$\begin{aligned}
&\mathrm{Tr}\left[\sigma_{x''} \cdot e^{-itH} \cdot \frac{\mathrm{Id} - \sigma_{x'}}{2^n} \cdot e^{itH}\right] \\
&= \mathrm{Tr}\left[\sigma_{x''} \cdot \left(\mathrm{Id}_{2^n} - itH + ct^2 R_1(t)\right) \cdot \frac{\mathrm{Id} - \sigma_{x'}}{2^n} \cdot \left(\mathrm{Id}_{2^n} + itH + ct^2 R_1(t)\right)\right] \\
&= \frac{1}{2^n}\Big(\mathrm{Tr}[\sigma_{x''}(\mathrm{Id} - \sigma_{x'})] - it\,\mathrm{Tr}[\sigma_{x''}H(\mathrm{Id} - \sigma_{x'})] + it\,\mathrm{Tr}[\sigma_{x''}(\mathrm{Id} - \sigma_{x'})H] + O(t^2)\,\mathrm{Tr}[\sigma_{x''}R]\Big) \\
&= \frac{1}{2^n}\Big(-it\,\mathrm{Tr}[\sigma_{x''}H(\mathrm{Id} - \sigma_{x'})] + it\,\mathrm{Tr}[\sigma_{x''}(\mathrm{Id} - \sigma_{x'})H]\Big) + O(t^2)\|H\|^2 \\
&= \frac{it}{2^n}(-\mathrm{Tr}[\sigma_{x''}H] + \mathrm{Tr}[\sigma_{x'}\sigma_{x''}H] + \mathrm{Tr}[\sigma_{x''}H] - \mathrm{Tr}[\sigma_{x''}\sigma_{x'}H]) + O(t^2)\|H\|^2 \\
&= \frac{2ita}{2^n}\,\mathrm{Tr}[\sigma_x H] + O(t^2)\|H\|^2 \\
&= 2ita\lambda_x + O(t^2)\|H\|^2 \qquad (17)
\end{aligned}$$

where in the first line used the Taylor expansion, third inequality used $\|R\|_{\mathrm{op}} \leq \|H\|$, $\mathrm{Tr}[\sigma_{x''}] = \mathrm{Tr}[\sigma_x] = 0$ and $\mathrm{Tr}[\sigma_{x''}R] \leq \|\sigma_{x''}\|_1\|R\|_{\mathrm{op}} \leq 2^n\|H\|$, and in the fifth inequality we used that

$$\sigma_{x'}\sigma_{x''} = -\sigma_{x''}\sigma_{x'} = a\sigma_x.$$

Thus, by taking $t = \varepsilon/\|H\|^2$ and dividing by $2i\varepsilon/\|H\|^2 a$ on both sides of Eq. (17), we have that

$$\left|\lambda_x - \frac{\|H\|^2}{2i\varepsilon a} \cdot \mathrm{Tr}\left[\sigma_{x''}e^{-i\varepsilon H}\frac{\mathrm{Id} - \sigma_{x'}}{2^n}e^{i\varepsilon H}\right]\right| = O(\varepsilon).$$

Hence, if we estimate $\mathrm{Tr}[\sigma_{x''}e^{-i\varepsilon/\|H\|^2 H}\frac{\mathrm{Id}-\sigma_{x'}}{2^n}e^{i\varepsilon/\|H\|^2 H}]$ up to error $\varepsilon^2/\|H\|^2$, which can be done with $O(\|H\|^4/\varepsilon^4)$ queries to $e^{-i\varepsilon/\|H\|^2 H}$, we obtain an ε-estimate of λ_x. □

3.4. Pauli Hashing. In this section, we describe how the n-qubit Pauli operators may be *hashed* or isolated into separate sets (which we will call *buckets*) based on their commutation relations with a subgroup of Pauli operators that will be defined shortly. We will then observe that given a function $f : \{0,1\}^{2n} \to [-1, 1]$ taking $2n$-bit strings corresponding to the Weyl operators to values in $[-1, 1]$, the hashing process gives us access to projections of these functions onto each of these buckets described by its symplectic Fourier transform. This entire hashing process can thus be viewed as *symplectic* Fourier hashing, analogous to Fourier hashing used by Gopalan et al. [GOS+11].

3.4.1. Hashing to cosets of a random subgroup We now describe our approach for pairwise independently hashing the Weyl operators (or the symplectic Fourier characters of f defined earlier). Consider a set of $2n$-bit strings $\beta_1, \ldots, \beta_t$ sampled independently and uniformly at random from $\mathbb{F}_2^{2n}$. We define H to be the subspace $H = \mathrm{span}\{\beta_1, \ldots, \beta_t\}$ spanned by these t vectors. The dimension of H is thus t and $|H| = 2^t$.[7] The set of Weyl

[7] We remark that the probability these t vectors are linearly independent is $\geq 1 - t2^{t-2n}$ and since the t we eventually pick will be $\ll 2n$, this probability is negligible and from here onwards for simplicity we will assume that these vectors are linearly independent.

operators corresponding to elements in H is a subgroup S of the n-qubit Pauli group and the Weyl operators corresponding to $\{\beta_j\}_{j\in[t]}$ are the generators of S. For each $b \in \mathbb{F}_2^t$, define the *bucket*

$$C(b) := \{\alpha \in \mathbb{F}_2^{2n} : [\alpha, \beta_j] = b_j \; \forall j \in [t]\}. \tag{18}$$

Let $\mathcal{C} = \{C(b)\}_{b\in\mathbb{F}_2^t}$ be the set of all buckets. Moreover, observe that $\{C(b)\}$ are the cosets in $\mathsf{A}(H) = \mathbb{F}_2^{2n}/\mathsf{C}(H)$, because $z \in \mathsf{C}(H)$ if and only if $[z, \beta_j] = 0$ for all $j \in [t]$. Also, note that $|\mathsf{C}(H)| = 2^{2n-t}$ and $|\mathsf{A}(H)| = 2^t$. We now argue that this indeed corresponds to a random hashing process, similar to [GOS+11, Proposition 2.9].

Proposition 3.7. *The following facts are true for the random coset structure described so far.*

(*i*) *For each* $\alpha \in \mathbb{F}_2^{2n} \setminus 0^{2n}$ *and each* b, *we have* $\Pr_H[\alpha \in C(b)] = 2^{-t}$.
(*ii*) *Consider distinct* $\alpha, \alpha' \in \mathbb{F}_2^{2n}$. *Then,* $\Pr_H[\alpha, \alpha' \in C(b)$ *for some* $b] = 2^{-t}$.
(*iii*) *Suppose* $S \subset \mathbb{F}_2^{2n}$ *with* $|S| \leq s + 1$. *Then,* $t \geq 2\log s + \log(3)$ *ensures all elements in* S *fall into different buckets with probability at least* $2/3$.
(*iv*) *For each* $b \in \mathbb{F}_2^t$ *and distinct* $\alpha, \alpha' \in \mathbb{F}_2^{2n}$, *we have* $\Pr_H[\alpha, \alpha' \in C(b)] = \Pr_H[\alpha \in C(b)] \cdot \Pr_H[\alpha' \in C(b)]$ *i.e., the random variables* $[\alpha \in C(b)]$ *and* $[\alpha' \in C(b)]$ *are independent.*[8]

Proof. (*i*) is true because as β_i is chosen uniformly at random, so for a fixed α, $\Pr[[\alpha, \beta_i] = 1] = 1/2$. (*ii*) follows by applying (*i*) to $\alpha - \alpha'$. (*iii*) follows from (*ii*) and a union bound over the $\leq s^2$ pairs of elements of S. To prove (*iv*) we divide it in two cases. In the first case, $\alpha = 0^n$. As 0^n always belongs to $C(0^t)$, the statement reads as $\Pr_H[\alpha' \in C(0^t)] = \Pr_H[\alpha' \in C(0^t)]$, which is true. If both $\alpha, \alpha' \neq 0^n$, then by (*i*) the RHS of (*iv*) is 4^{-t}. The LHS is 4^{-t} too because

$$2^{-t} \underbrace{=}_{(ii)} \Pr_H[\alpha, \alpha' \in C(b) \text{ for some } b] = \sum_{b\in\{0,1\}^t} \Pr_H[\alpha, \alpha' \in C(b)]$$

and $\Pr_H[\alpha, \alpha' \in C(b)]$ does not depend on b. □

3.4.2. Projection of functions onto cosets We now describe the functions corresponding to the different cosets obtained during hashing. Let $V \subseteq \mathbb{F}_2^{2n}$ be a subspace. Consider the symplectic complement of V, denoted by $\mathsf{C}(V) = \{z \in \mathbb{F}_2^{2n} : \forall v \in V,\ [z, v] = 0\}$. For $a \in \mathbb{F}_2^{2n}$, define the coset $a + V := \{a + v : v \in V\}$. Given a function $f : \mathbb{F}_2^{2n} \to [-1, 1]$, we define the "projection function" $f|_{a+V}$ as

$$f|_{a+V} = \sum_{\beta\in a+V} \breve{f}(\beta)\chi_\beta(z),$$

where $\chi_\beta(x) = (-1)^{[\beta,x]}$. Notably, the symplectic Fourier coefficients of the projected function are

$$\breve{f}\,|_{a+V}(\alpha) = \begin{cases} \breve{f}(\alpha) & \text{if } \alpha \in a + V, \\ 0 & \text{otherwise.} \end{cases}$$

[8] We remark that 0^{2n} will always lie in bucket corresponding to $\mathsf{C}(0^t)$. We can randomize this by further adding a random permutation after the construction of the cosets as done in [GOS+11].

Fact 3.8. *The projected function $f|_{a+V}$ satisfies $f|_{a+V}(z) = \mathbb{E}_{x\in\mathsf{C}(V)}[f(x+z)\chi_a(x)]$.*

Proof. Expanding the right hand side, we have

$$\begin{aligned}
\mathbb{E}_{x\in\mathsf{C}(V)}[f(x+z)\chi_a(x)] &= \frac{1}{|\mathsf{C}(V)|}\sum_{x\in\mathsf{C}(V)}\left[\sum_{b\in\mathbb{F}_2^{2n}}\breve{f}(b)(-1)^{[b,x+z]}\right](-1)^{[a,x]} \\
&= \frac{1}{|\mathsf{C}(V)|}\sum_{b\in\mathbb{F}_2^{2n}}\breve{f}(b)(-1)^{[b,z]}\left[\sum_{x\in\mathsf{C}(V)}(-1)^{[a+b,x]}\right] \\
&= \frac{1}{|\mathsf{C}(V)|}\sum_{b\in\mathbb{F}_2^{2n}}\breve{f}(b)(-1)^{[b,z]}|\mathsf{C}(V)|[a+b\in V] \\
&= \sum_{b\in\mathbb{F}_2^{2n}}\breve{f}(b)(-1)^{[b,z]}[b\in a+V] \\
&= \sum_{b\in a+V}\breve{f}(b)(-1)^{[b,z]} \\
&= f|_{a+V}(z),
\end{aligned}$$

where we expanded $f(x+z)$ by considering its symplectic Fourier expansion in the first equality, used Fact 2.6 in the fourth equality and the last equality follows from the definition of the projected function. This completes the proof. □

We define the weight $\mathsf{wt}_f(a+V) = \sum_{\alpha\in a+V}\breve{f}(\alpha)^2$.

Fact 3.9. *The weight of the function f on a coset $a+V$ can be evaluated as*

$$\mathsf{wt}_f(a+V) = \mathop{\mathbb{E}}_{\substack{x\in\mathbb{F}_2^{2n},\\ z\in\mathsf{C}(V)}}[\chi_a(z)f(x)f(x+z)].$$

This can be estimated up to ε-error with probability $1-\delta$ using $O(1/\varepsilon^2\log(1/\delta))$ queries to f.

Proof. Using Parseval's theorem and Fact 3.8, we have that

$$\begin{aligned}
\mathsf{wt}_f(a+V) = \mathbb{E}_{w\in\mathbb{F}_2^{2n}}\left[f|_{a+V}(w)^2\right] &= \mathbb{E}_{w\in\mathbb{F}_2^{2n}}\mathbb{E}_{y_1,y_2\in\mathsf{C}(V)}[f(y_1+w)f(y_2+w)\chi_a(y_1+y_2)] \\
&= \mathbb{E}_{y_1,y_2\in\mathsf{C}(V)}\mathbb{E}_{x\in\mathbb{F}_2^{2n}}[f(x)f(y_1+y_2+x)\chi_a(y_1+y_2)] \\
&= \mathbb{E}_{z\in\mathsf{C}(V)}\mathbb{E}_{x\in\mathbb{F}_2^{2n}}[f(x)f(x+z)\chi_a(z)],
\end{aligned}$$

where in the second line we rewrote $x = y_1+w$, and the third line we rewrote $z = y_1+y_2$ and used that $\mathsf{C}(V)$ is a subspace. Estimating the weight on $a+V$ requires queries to f. The query complexity follows from the Hoeffding bound (Lemma 2.4) and that the term inside the expectation lies in $[-1,1]$. □

4. Testing and Learning with Quantum Memory

In this section, we give our testing and learning algorithms for local Hamiltonians.

4.1. Testing local Hamiltonians. We now state our locality testing algorithm and prove its guarantees.

Algorithm 4 Locality tester

Input: Query access to the time evolution of $U(t) = e^{-itH}$, closeness and farness parameters $\varepsilon_1,\ \varepsilon_2 \in (0, 1)$, locality parameter $k \in \mathbb{N}$ and failure parameter $\delta \in (0, 1)$.

1: Set $T = O(\log(1/\delta)\|H\|^4/(\varepsilon_2 - \varepsilon_1)^4)$
2: Let $t = (\varepsilon_2 - \varepsilon_1)/(3c\|H\|^2)$ and $U = U(t)$
3: Initialize $\alpha'_k = 0$
4: **for** $i = 1, ..., T$ **do**
5: Perform Pauli sampling from U. Let $x \in \{0, 1, 2, 3\}^n$ be the outcome.
6: **if** $|x| > k$ **then**
7: $\alpha'_k \leftarrow \alpha'_k + 1/T$
8: IInitialize $\alpha''_k = 0$
9: **for** $i = 1, ..., T$ **do**
10: Perform Pauli sampling from U. Let $x \in \{0, 1, 2, 3\}^n$ be the outcome.
11: If $|x| > k$, $\alpha''_k \leftarrow \alpha''_k + 1/T$

Output: If $\alpha'_k \geq (3/4)(\varepsilon_2-\varepsilon_1)^2/\|H\|^2$ or $\alpha''_k \geq ((\varepsilon_2-\varepsilon_1)(\varepsilon_1+2\varepsilon_2)/(9c\|H\|^2))^2-((\varepsilon_2-\varepsilon_1)^2/(18c\|H\|^2))^2$ output that H is far from local, and close to local otherwise

Note: Steps 4 to 7 of the algorithm are the same as steps 8 to 11 and are unnecessary. The algorithm would have the same guarantees if steps 4 to 7 were not performed, and one would output that H is far from local only when $\alpha''_k \geq ((\varepsilon_2 - \varepsilon_1)(\varepsilon_1 + 2\varepsilon_2)/(9c\|H\|^2))^2 - ((\varepsilon_2 - \varepsilon_1)^2/(18c\|H\|^2))^2$. However, we have included this redundancy in order to have a cleaner analysis of the performance guarantees of the algorithm in the proof of Theorem 4.1.

Theorem 4.1. *Algorithm 4 solves the locality testing problem (Problem 1.1 with the property of being k-local) with probability* $\geq 1 - \delta$*, by making* $O(\|H\|^4/(\varepsilon_2 - \varepsilon_1)^4 \cdot \log(1/\delta))$ *queries to the evolution operator and with* $O(\|H\|^2/(\varepsilon_2 - \varepsilon_1)^3 \cdot \log(1/\delta))$ *total evolution time.*

Proof. Let $t = (\varepsilon_2 - \varepsilon_1)/(3c\|H\|^2)$ and let $U = U(t)$. For notational simplicity, let $\alpha_k := \|U_{>k}\|_2^2$. We will first estimate α_k upto error $(\varepsilon_2 - \varepsilon_1)^2/(4\|H\|^2)$. To do that we sample from $\{|\widehat{U}_x|^2\}_x$ using Fact 2.1 a total of $T = O(\|H\|^4/(\varepsilon_2-\varepsilon_1)^4 \log(1/\delta))$ times, which can be done with T queries. If $x_1, \ldots, x_T$ are the outcomes of those samples, we define our estimate as

$$\alpha'_k := \frac{1}{T} \sum_{i\in[T]} [|x_i| > k].$$

By the Hoeffding bound, we have that indeed $|\alpha'_k - \alpha_k| \leq (\varepsilon_2 - \varepsilon_1)^2/(4\|H\|^2)$ with probability $\geq 1 - \delta/2$.

If $\alpha'_k \geq (3/4)(\varepsilon_2 - \varepsilon_1)^2/\|H\|^2$, then $\alpha_k \geq (\varepsilon_2 - \varepsilon_1)^2/(2\|H\|^2)$, so by Lemma 3.1 we conclude that H is far from k-local. Otherwise, if $\alpha'_k \leq (3/4)(\varepsilon_2 - \varepsilon_1)^2/\|H\|^2$, then $\alpha_k \leq (\varepsilon_2 - \varepsilon_1)^2/\|H\|^2$. Now we take again T samples from $y_1, \ldots, y_T$ from $\{|\widehat{U}_x|^2\}_x$ and define a new estimate

$$\alpha''_k = \frac{1}{T} \sum_{i\in[T]} [|y_i| > k].$$

By definition α_k'' equals α_k in expectation. Furthermore, α_k is the empirical average of random variables whose variance is considerably small, because

$$\mathbb{E}[[|y| > k]^2] = \mathbb{E}[[|y| > k]] = \|U_{>k}\|_2^2 \leq \frac{(\varepsilon_2 - \varepsilon_1)^2}{\|H\|^2}.$$

Then, an application of Bernstein's inequality (Lemma 2.5) shows that α_k'' approximates $\|U_{>k}\|_2^2$ up to error $((\varepsilon_2 - \varepsilon_1)^2/(18c\|H\|^2))^2$ with success probability $1 - \delta/2$. At this point, using our structure Lemma 3.1, this is sufficient for testing k-locality. □

We also note that the algorithm for testing locality can be used in more generality for testing if the support of the Hamiltonians is a given $\mathcal{S} \subseteq \{0, 1, 2, 3\}^n$. Also, by a union bound one can test for M supports $\mathcal{S}_1, \ldots, \mathcal{S}_M$ by paying a factor $\log(M)$.

Theorem 4.2. *Let H be a n-qubit Hamiltonian and let $\mathcal{S}_1, \ldots, \mathcal{S}_M \subseteq \{0, 1, 2, 3\}^n$. Then, with $O(\|H\|^4/(\varepsilon_2 - \varepsilon_1)^4 \log(M/\delta))$ queries and $O(\|H\|^2/(\varepsilon_2 - \varepsilon_1)^3 \log(M/\delta))$ total evolution time one can simultaneously, for every $i \in [M]$, test if H is ε_1-close to or ε_2-far from being supported on $\mathcal{S}_i$.*

Theorem 4.1 is one case of Theorem 4.2 where $M = 1$ and $\mathcal{S}_1 = \{x \in \{0, 1, 2, 3\}^n : |x| \leq k\}$.

4.2. Testing sparse Hamiltonians. Now we state our sparsity testing algorithm and prove its guarantees.

Algorithm 5 Fully tolerant sparsity tester

Input: Query access to the time evolution of $U(t) = e^{-itH}$, closeness and farness parameters $\varepsilon_1,\ \varepsilon_2 \in (0, 1)$, sparsity parameter $s \in \mathbb{N}$ and failure parameter $\delta \in (0, 1)$

1: Set $T = O(s^6\|H\|^{12}/(\varepsilon_2^2 - \varepsilon_1^2)^6 \cdot \log(1/\delta))$

2: Let $t = O((\varepsilon_2^2 - \varepsilon_1^2)/s\|H\|^3)$ and $U = U(t)$

3: Perform Pauli sampling from U a total of T times. Let $(|\alpha_x|^2)_{x\in\{0,1,2,3\}^n}$ the empirical estimate of $(|\widehat{U}_x|^2)_x$ obtained this way.

4: Let $|\alpha_{x_1}|^2, \ldots, |\alpha_{x_s}|^2$ the s-biggest elements of $(|\alpha_x|^2)_{x\in\{0,1,2,3\}^n - \{0^n\}}$

5: Set $\Gamma = |\alpha_{0^n}|^2 + \sum_{i\in[s]} |\alpha_{x_i}|^2$.

Output: If $\Gamma \geq 1 - \varepsilon_1^2 \frac{(\varepsilon_2^2-\varepsilon_1^2)^2}{s^2\|H\|^6} - \frac{1}{2}\frac{(\varepsilon_2^2-\varepsilon_1^2)^3}{s^2\|H\|^6}$ output that H is close to sparse, and far from sparse otherwise

Theorem 4.3. *Algorithm 5 solves the s-sparsity testing problem with probability $\geq 1-\delta$, by making $O(s^6\|H\|^{12}/(\varepsilon_2^2 - \varepsilon_1^2)^6 \cdot \log(1/\delta))$ queries to the evolution operator and with $O(s^5\|H\|^9/(\varepsilon_2^2 - \varepsilon_1^2)^5 \cdot \log(1/\delta))$ total evolution time.*

Proof. Let $t = O((\varepsilon_2^2 - \varepsilon_1^2)/s\|H\|^3)$, which satisfies $t \leq 1/(2\|H\|_{\text{op}})$. By Lemma 3.2 we have that if H is ε_1-close to being sparse, then

$$\mathsf{TopEnergy}(t; s) \geq 1 - \varepsilon_1^2 \frac{(\varepsilon_2^2 - \varepsilon_1^2)^2}{s^2\|H\|^6} - \frac{1}{3}\frac{(\varepsilon_2^2 - \varepsilon_1^2)^3}{s^2\|H\|^6},$$

while if H is ε_2-far from s-sparse, then

$$\mathsf{TopEnergy}(t;s) \le 1 - \varepsilon_2^2 \frac{(\varepsilon_2^2-\varepsilon_1^2)^2}{s^2\|H\|^6} + \frac{1}{3}\frac{(\varepsilon_2^2-\varepsilon_1^2)^3}{s^2\|H\|^6}.$$

From here, it follows that to test for sparsity, it suffices to estimate $\mathsf{TopEnergy}(t;s)$ up to error

$$\begin{aligned}\varepsilon &= \frac{1}{2}\left(1-\varepsilon_1^2\frac{(\varepsilon_2^2-\varepsilon_1^2)^2}{s^2\|H\|^6} - \frac{1}{3}\frac{(\varepsilon_2^2-\varepsilon_1^2)^3}{s^2\|H\|^6} - \left\{1-\varepsilon_2^2\frac{(\varepsilon_2^2-\varepsilon_1^2)^2}{s^2\|H\|^6} + \frac{1}{3}\frac{(\varepsilon_2^2-\varepsilon_1^2)^3}{s^2\|H\|^6}\right\}\right)\\ &= \frac{(\varepsilon_2^2-\varepsilon_1^2)^3}{6s^2\|H\|^6}.\end{aligned}$$

To do that we will obtain an estimate $(\{|\alpha_x|^2\}_x$ of $\{|\widehat{U}_x|^2\}_x$ and use it to approximate $\mathsf{TopEnergy}(t;s)$. Using Fact 2.2, we obtain an empirical distribution $\{|\alpha_x|^2\}_x$ that is obtained after $T = O(s^2\log(1/\delta)/\varepsilon^2)$ samples from $\{|\widehat{U}_x|^2\}_x$ (which can be performed with T queries to $U(t)$ thanks to Fact 2.1) satisfies that

$$\left||\alpha_x|^2 - |\widehat{U}_x|^2\right| \le \frac{\varepsilon}{2s+1} \tag{19}$$

for all $x \in \{0,1\}^{2n}$ with probability $\ge 1-\delta$. We assign new labels $y_0, y_1, \ldots, y_{4^n-1}$ to $\{0,1\}^{2n}$ in a way such that $|\alpha_{y_0}|^2 = |\alpha_{0^n}|^2$ and $|\alpha_{y_1}|^2 \ge \cdots \ge |\alpha_{y_{4^n-1}}|^2$. Now, we define our estimate for $\mathsf{TopEnergy}(t;s)$ as

$$\mathsf{TopEnergy}'(t;s) = |\alpha_{y_0}(t)|^2 + 2\sum_{i\in[s]}|\alpha_{y_i}(t)|^2.$$

It only remains to show that $\mathsf{TopEnergy}'(t;s)$ ε-approximates $\mathsf{TopEnergy}(t;s)$. We will see that in two steps. First,

$$\begin{aligned}\mathsf{TopEnergy}'(t;s) &= |\alpha_{y_0}(t)|^2 + 2\sum_{i\in[s]}|\alpha_{y_i}(t)|^2\\ &\ge |\alpha_{x_0}(t)|^2 + 2\sum_{i\in[s]}|\alpha_{x_i}(t)|^2\\ &\ge |u_{x_0}(t)|^2 + 2\sum_{i\in[s]}|u_{x_i}(t)|^2 - \varepsilon\\ &= \mathsf{TopEnergy}(t;s) - \varepsilon,\end{aligned}$$

where the second line is true by definition of $y_0, \ldots, y_{4^n-1}$ and the third line is true because Eq. (19). Switching the roles of $\mathsf{TopEnergy}'(t;s)$ and $\mathsf{TopEnergy}(t;s)$, one can prove that $\mathsf{TopEnergy}(t;s) \ge \mathsf{TopEnergy}'(t;s) - \varepsilon$.

Complexity analysis. We have queried $U(t)$ a total of $T = O(s^2\log(1/\delta)/\varepsilon^2)$ times with $\varepsilon = (\varepsilon_2^2-\varepsilon_1^2)^3/(6s^2\|H\|^6)$ and $t = O((\varepsilon_2^2-\varepsilon_1^2)/(s\|H\|^3))$, so the number of queries is

$$O\left(\frac{s^6\|H\|^{12}}{(\varepsilon_2^2-\varepsilon_1^2)^6}\log(1/\delta)\right)$$

and the total evolution time

$$O\left(\frac{s^5\|H\|^9}{(\varepsilon_2^2-\varepsilon_1^2)^5}\log(1/\delta)\right).$$

□

Furthermore, for the regime where $\varepsilon_1 = O(\varepsilon_2/s^{0.5})$ we propose a more efficient testing algorithm.

Algorithm 6 Less tolerant sparsity tester

Input: Query access to the time evolution of $U(t) = e^{-itH}$, sparsity parameter $s \in \mathbb{N}$, closeness and farness parameters $\varepsilon_1,\ \varepsilon_2 \in (0,1)$ satisfying $\varepsilon_1 = O(\varepsilon_2/\sqrt{s})$ and failure parameter $\delta \in (0,1)$

1: Set $T = O(\|H\|^4 s^2/\varepsilon_2^4 \cdot \log(1/\delta))$

2: Let $t = \Omega(\varepsilon_2/(\sqrt{s}\|H\|^2))$ and $U = U(t)$

3: Perform Pauli sampling from U a total of T times. Let $\mathcal{X}$ the set of sampled Paulis.

Output: If $|\mathcal{X} - \{0^{2n}\}| \leq s$ output that H is close to sparse, and far from sparse otherwise

Theorem 4.4. *Algorithm 6 solves the s-sparsity testing problem with probability* $\geq 1-\delta$ *for* $\varepsilon_1 = O(\varepsilon_2/s^{0.5})$. *The algorithm makes* $O(s^2\|H\|^4/\varepsilon_2^4 \cdot \log(1/\delta))$ *queries to the evolution operator and uses* $O(s^{1.5}\|H\|^2/\varepsilon_2^3 \cdot \log(1/\delta))$ *total evolution time.*

Proof. By the first order Taylor expansion, Eq. (1), we have that

$$U(t) = \mathrm{Id} - itH + ct^2 R_1(t)\|H\|_{\mathrm{op}}$$

with $\|R_1\|_{\mathrm{op}}^2 \leq 1$ for

$$t \leq 1/2\|H\|_{\mathrm{op}}. \tag{20}$$

We will assume that $\delta = 1/3$, as the case $\delta \in (0, 1/3)$ follows by a standard majority voting argument.

Algorithm 6 is simple. One just performs Pauli sampling of $U = U(t)$ a number of T times, for some t and T to be determined later. Let $\mathcal{X}$ be the labels of the Pauli strings sampled in this process. If $|\mathcal{X} - \{0^{2n}\}| \leq s$ we output that H is sparse, and otherwise we output that is far from sparse. It remains to analyze the correctness.

Correctness. In the case that H is ε_1-close to s-sparse, there exists $\mathcal{S} \subseteq \{0,1\}^{2n}$ of size s where H is ε_1-concentrated. Then, by Taylor expansion,

$$\sqrt{\sum_{x\notin(\mathcal{S}\cup\{0^{2n}\})}|\widehat{U}_x|^2} \leq t\sqrt{\sum_{x\notin(\mathcal{S}\cup\{0^{2n}\})}|\lambda_x|^2} + ct^2\|H\|_{\mathrm{op}}^2 \leq t\varepsilon_1 + ct^2\|H\|_{\mathrm{op}}^2 \leq 2ct^2\|H\|^2,$$

where in the last inequality we have assumed that

$$\varepsilon_1 \leq ct\|H\|^2. \tag{21}$$

Hence, the probability of sampling an element outside $\mathcal{S}\cup\{0^{2n}\}$ in one sample is at most $4c^2t^4\|H\|^4$. Thus, the probability of not sampling an element outside $\mathcal{S}\cup\{0^{2n}\}$ in T samples is at least

$$(1-4c^2t^4\|H\|^4)^T \geq 1 - 4c^2t^4\|H\|^4 T.$$

In particular, if

$$T \leq \frac{1}{3}\frac{1}{4c^2t^4\|H\|^4} \tag{22}$$

it will be satisfied that $|\mathcal{X} - \{0^{2^n}\}| \leq s$ with probability $\geq 2/3$, as desired.

In the case that H is ε_2-far from s-sparse, we will perform an analysis similar to the coupon collector problem. By Taylor expansion, we have that for every set $\mathcal{S}$ of size s,

$$\sqrt{\sum_{x\notin(\mathcal{S}-\{0^{2n}\})} |\widehat{U}_x|^2} \geq \varepsilon_2 t - ct^2\|H\|_{\text{op}}^2 \geq \frac{\varepsilon_2 t}{2}, \tag{23}$$

where we have assumed that

$$\|H\|_{\text{op}}^2 ct \leq \varepsilon_2/2. \tag{24}$$

Let X_i be the random variable that accounts for the number of samples between the $(i-1)$-th sampled non-0^{2n}-Pauli and the i-th sampled non-0^{2n}-Pauli. Applying Eq. (23) to every $\mathcal{X}_i$, it follows that $\mathbb{E}[X_i] \leq 4/\varepsilon_2^2 t^2$ for every $i \in [s+1]$, so

$$\mathbb{E}[X_1 + \cdots + X_{s+1}] \leq \frac{4(s+1)}{\varepsilon_2^2 t^2}.$$

Hence, by Markov's inequality, if

$$T \geq \frac{\sqrt{3}4(s+1)}{\varepsilon_2^2 t^2} \tag{25}$$

it will be satisfied that $|\mathcal{X} - \{0^{2n}\}| \geq s+1$ with probability $\geq 2/3$, as desired.

Finally, we note that we have assumed conditions Eqs. (20) to (22),(24) and (25) to ensure the correctness of the algorithm. All these equations are satisfied provided that

$$\begin{aligned}
t &= \frac{\varepsilon_2}{\|H\|^2\sqrt{48\sqrt{3}c^2(s+1)}} = \Omega\left(\frac{\varepsilon_2}{\sqrt{s}\|H\|^2}\right),\\
T &= \frac{1}{12c^2t^4\|H\|^4} = O\left(\frac{s^2\|H\|^4}{\varepsilon_2^4}\right),\\
\varepsilon_1 &\leq \frac{\varepsilon_2}{\sqrt{48\sqrt{3}(s+1)}} = O\left(\frac{\varepsilon_2}{\sqrt{s}}\right).
\end{aligned}$$

□

Algorithm 7 Estimating a given set of Pauli coefficients of a Hamiltonian

Input: Query access to the time evolution of $U(t) = e^{-itH}$, target set of Pauli coefficients $\mathcal{X} \subseteq \{0,1\}^{2n} - \{0^n\}$, error parameter $\varepsilon \in (0,1)$, and failure parameter $\delta \in (0,1)$

1: Set $T = O(\|H\|^4/\varepsilon^4 \cdot \log(|\mathcal{X}|/\delta))$ and $t_0 = \Theta(\varepsilon/\|H\|^2)$
2: Set $U = U(t_0)$
3: **for** $j \in [T]$ **do**
4: Prepare $|J(U)\rangle = (U \otimes \mathrm{Id}_{2^n})|\mathrm{EPR}_n\rangle$
5: Apply a uniformly random Clifford gate C
6: Measure in the computational basis. Let $|b_j\rangle$ be the outcome
7: **for** $x \in \mathcal{X}$ **do**
8: Let $\mathcal{R}_{x,j} = (2^n+1)\langle b_j|C^{-1}\mathcal{R}_x C|b_j\rangle$ and $\mathcal{I}_{x,j} = (2^n+1)\langle b_j|C^{-1}\mathcal{I}_x C|b_j\rangle$
9: **for** $x \in \mathcal{X}$ **do**
10: Set $\widetilde{R}_x := \mathrm{MedianOfMeans}(\mathcal{R}_{x,j})_j$ and $\widetilde{I}_x := \mathrm{MedianOfMeans}(\mathcal{I}_{x,j})_j$

Output: $((\widetilde{R}_x + i\widetilde{I}_x)/(-it))_{x\in\mathcal{X}}$

4.3. Learning unstructured and structured Hamiltonians. We start by showing how to efficiently learn an arbitrary n-qubit Hamiltonian in ℓ_∞ error. To do that, we propose a protocol to estimate a given set of Pauli coefficients $\mathcal{X}$ of a Hamiltonian via Shadow tomography. To describe the protocol, we introduce the following $2n$-qubit observables. Given $x \in \{0,1\}^{2n}$, we define

$$\mathcal{R}_x := \frac{1}{2}(|\mathrm{Bell}_{0^{2n}}\rangle\langle\mathrm{Bell}_x| + |\mathrm{Bell}_x\rangle\langle\mathrm{Bell}_{0^{2n}}|),$$
$$\mathcal{I}_x := \frac{1}{2}(-i|\mathrm{Bell}_{0^{2n}}\rangle\langle\mathrm{Bell}_x| + i|\mathrm{Bell}_x\rangle\langle\mathrm{Bell}_{0^{2n}}|),$$

where $|\mathrm{Bell}_x\rangle := (\mathrm{Id}\otimes\sigma_x)|\mathrm{EPR}_n\rangle$.

Lemma 4.5. *Let H be an n-qubit traceless Hamiltonian and $\mathcal{X} \subseteq \{0,1\}^{2n}$. Then, Algorithm 7 allows to estimate the Pauli coefficients corresponding to $\mathcal{X}$ with success probability $\geq 1-\delta$. It uses $O((\log|\mathcal{X}|/\delta)\|H\|^4/\varepsilon^4)$ queries and $O(\log(|\mathcal{X}|/\delta)\|H\|^2/\varepsilon^3)$ total evolution time.*

The minimum evolution time is $\varepsilon/\|H\|^2$, the number of ancillas is n, and the time complexity is $O(\mathrm{poly}(n)|\mathcal{X}|\|H\|^4/\varepsilon^4 \cdot \log(|\mathcal{X}|/\delta))$.

Proof. **Correctness of the algorithm:** Let $t_0 = \Theta(\varepsilon/\|H\|^2)$ and $U = U(t_0)$. As $\mathrm{Tr}[\mathcal{R}_x^2] = \mathrm{Tr}[\mathcal{I}_x^2] = 2$, by Theorem 2.3, the numbers $\widetilde{R}_x$ and $\widetilde{I}_x$ that Algorithm 7 outputs satisfy

$$|\mathrm{Tr}[\mathcal{R}_x|J(U)\rangle\langle J(U)|] - \widetilde{R}_x| \leq \frac{\varepsilon^2}{\|H\|^2}, \quad |\mathrm{Tr}[\mathcal{I}_x|J(U)\rangle\langle J(U)|] - \widetilde{I}_x| \leq \frac{\varepsilon^2}{\|H\|^2}, \tag{26}$$

for every $x \in \mathcal{X}$ with probability $\geq 1-\delta$. By Taylor expansion, as $\lambda_{0^{2n}} = 0$, we have that $|\widehat{U}_{0^{2n}} - 1| \leq O(t_0^2\|H\|^2)$. Thus,

$$\begin{aligned}\mathrm{Tr}[\mathcal{R}_x|J(U)\rangle\langle J(U)|] &= \frac{1}{2}(\widehat{U}_x\widehat{U}_{0^{2n}}^* + \widehat{U}_{0^{2n}}\widehat{U}_x^*) \\ &= \mathrm{Re}(\widehat{U}_x\widehat{U}_0^*) = \mathrm{Re}(\widehat{U}_x) \pm O(t_0^2\|H\|^2),\end{aligned} \tag{27}$$

and similarly $\mathrm{Tr}[\mathcal{I}_x|J(U)\rangle\langle J(U)|] = \mathrm{Im}(\widehat{U}_x) \pm O(t_0^2\|H\|^2)$. Hence, combining Eqs. (26) and (27) we have that

$$|\widehat{U}_x - (\widetilde{R}_x + i\widetilde{I}_x)| \leq \frac{\varepsilon^2}{\|H\|^2} + O(t_0^2\|H\|^2) \leq O\left(\frac{\varepsilon^2}{\|H\|^2}\right),$$

for every $x \in \mathcal{X}$. Finally, by Taylor expansion we have that $|\widehat{U}_x/(-it_0) - \lambda_x| \leq O(t_0\|H\|^2)$, so

$$\left|\lambda_x - \frac{\widetilde{R}_x + i\widetilde{I}_x}{-it_0}\right| \leq O\left(\frac{\varepsilon^2}{t_0\|H\|^2}\right) + O(t_0\|H\|^2) = O(\varepsilon),$$

for every $x \in \mathcal{X}$, as claimed.

Time complexity: The time complexity is dominated by the first loop in Algorithm 7, whose time complexity is $O(|\mathcal{X}| \cdot T \cdot (t_{est} + \mathrm{poly}(n))$, where the $\mathrm{poly}(n)$ comes from applying a random Clifford gate and t_{est} is the time taken to compute $\langle b|C^{-1}\mathcal{R}_x C|b\rangle$ for an n-qubit Clifford gate C and a computational basis state $|b\rangle$. Now, expanding R_x one can write $\langle b|C^{-1}\mathcal{R}_x C|b\rangle$ as an algebraic expression of a finite number of terms of the kind $\langle y|D|z\rangle$, where $|y\rangle$ and $|z\rangle$ are computational basis states and D a Clifford gate. Hence, via Gottesman-Knill theorem [Got98, AG04] follows that $t_{est} = O(n^2)$, so the total time complexity is $O(\mathrm{poly}(n)|\mathcal{X}|\|H\|^4/\varepsilon^4 \cdot \log(|\mathcal{X}|/\delta))$. □

Now, we are ready to present our learning algorithm for arbitrary Hamiltonians with no promise about its structure.

Algorithm 8 Learning unstructured Hamiltonians

Input: Query access to the time evolution of $U(t) = e^{-itH}$, error parameter $\varepsilon \in (0, 1)$, and failure parameter $\delta \in (0, 1)$

1: Set $T = O(\|H\|^4/\varepsilon^4 \cdot \log(\|H\|^2/\varepsilon^2\delta))$ and $t_0 = \Theta(\varepsilon/\|H\|^2)$
2: Set $U = U(t_0)$
3: Set $\mathcal{X} = \emptyset$
4: **for** $j \in [T]$ **do**
5: Prepare $|J(U)\rangle = (U \otimes \mathrm{Id}_{2^n})|\mathrm{EPR}_n\rangle$
6: Measure in the Bell basis and add the outcome $x \in \{0, 1\}^{2n}$ to $\mathcal{X}$ if $x \neq 0^{2^n}$
7: Run Algorithm 7 run with $U(t)$, $\mathcal{X}$, ε and δ as inputs. Let $(\widetilde{\lambda}_x)_{x\in\mathcal{X}}$ the output.

Output: $\widetilde{H} = \sum_{x\in\mathcal{X}} \widetilde{\lambda}_x \sigma_x$

Theorem 4.6 (Learning unstructured Hamiltonians). *Let H be an n-qubit and traceless Hamiltonian. Then, Algorithm 8 ε-learns H in the ℓ_∞ norm with success probability $\geq 1 - \delta$. It uses $\widetilde{O}((\|H\|/\varepsilon)^4)$ queries to the evolution operator and $\widetilde{O}(\|H\|^2/\varepsilon^3)$ total evolution time. The minimum evolution time is $\Theta(\varepsilon/\|H\|^2)$, the algorithm uses n ancilla qubits and only one round of adaptivity, and the time complexity is* $\mathrm{poly}(n, 1/\varepsilon, \|H\|)$.

Proof. Let $t_0 = \Theta(\varepsilon/\|H\|^2)$, $U = U(t_0)$ and $T = O(\|H\|^4/\varepsilon^4 \cdot \log(\|H\|^2/\varepsilon^2\delta))$, as in Algorithm 8.

Correctness: We claim that with probability $\geq 1-\delta$ the set $\mathcal{X}$ generated in Algorithm 8 contains all x such that

$$|\lambda_x| \geq \varepsilon, \tag{28}$$

and that

$$|\mathcal{X}| \leq \widetilde{O}\left(\frac{\|H\|^4}{\varepsilon^4}\right). \tag{29}$$

To show Eq. (28) we note that by Taylor expansion, if $|\lambda_x| \geq \varepsilon$, then $|\widehat{U}_x| = \Omega((\varepsilon^2/\|H\|^2))$, so $|\widehat{U}_x|^2 = \Omega((\varepsilon^4/\|H\|^4))$. Hence, the probability that such an x does not belong to $\mathcal{X}$, which stores the non-0^{2n} outcomes of sampling from $(|\widehat{U}_x|^2)_x$, is at most

$$\left(1 - |\widehat{U}_x|^2\right)^T \leq e^{-T|\widehat{U}_x|^2} \leq \frac{\varepsilon^2\delta}{\|H\|^2}.$$

Hence, as there is at most $\|H\|^2/\varepsilon^2$ coefficients with $|\lambda_x| \geq \varepsilon$, because $\sum_x |\lambda_x|^2 \leq \|H\|^2$, Eq. (28) follows from a union bound. Eq. (29) holds because $|\mathcal{X}| \leq T$.

Now, if Eqs. (28) and (29) are satisfied, Algorithm 7 provides estimates of the coefficients of $\mathcal{X}$, which contains all labels x of coefficients $|\lambda_x| \geq \varepsilon$.

Complexities: The query complexity is $2T = \widetilde{O}(\|H\|^4/\varepsilon^4)$, the minimum evolution time $t_0 = \Theta(\varepsilon/\|H\|^2)$ and the total time evolution $2Tt_0 = \widetilde{O}(\|H\|^2/\varepsilon^3)$. Additionally, the time complexity of Algorithm 8 is dominated by the call to Algorithm 7, which runs in time $O(\text{poly}(n)|\mathcal{X}|\|H\|^2/\varepsilon^2)$, which thanks to Eq. (29) is $\text{poly}(n, 1/\varepsilon, \|H\|)$. □

4.3.1. Learning local Hamiltonians We now introduce our local Hamiltonian learner and prove its guarantees.

Algorithm 9 Local Hamiltonian learner

Input: Query access to the time evolution of $U(t) = e^{-itH}$, error parameter $\varepsilon \in (0, 1)$, locality parameter $k \in \mathbb{N}$ and failure parameter $\delta \in (0, 1)$

1: Set $T = \exp(O(k^2 + k\log(1/\varepsilon))\log(1/\delta)$
2: Let $t = \varepsilon^{k+1}\exp(-k(k+1)/2)$ and $U = U(t)$
3: Set $\gamma = (\varepsilon/\|H\|^2)^{k+1}\exp(-k(k+1)/2)$ and $\beta = \gamma\varepsilon/\|H\|$
4: Learn β-estimates λ'_x of λ_x via Algorithm 8
5: **for** $|x| \leq k$ **do**
6: **if** $|\lambda'_x| \leq \gamma$ **then**
7: $\widetilde{\lambda}_x = 0$
8: **else**
9: $\widetilde{\lambda}_x = \lambda'_x$

Output: $\sum_{x\leq k} \widetilde{\lambda}_x\sigma_x$

Theorem 4.7. *Given a n-qubit k-local Hamiltonian H, Algorithm 9 outputs $\widetilde{H}$ such that with probability $\geq 1 - \delta$ satisfies $\|H - \widetilde{H}\|_{\ell_2} \leq \varepsilon$. The algorithm makes $\exp(O(k^2 + k\log(\|H\|^2/\varepsilon))\log(1/\delta)$ queries to the evolution operator with $\exp(O(k^2 + k\log(\|H\|^2/\varepsilon))\log(1/\delta)$ total evolution time.*

To prove this theorem, we use the non-commutative Bohnenblust-Hille inequality [VZ23].

Theorem 4.8 (Non-Commutative Bohnenblust–Hille inequality). *Let $H = \sum_x \lambda_x\sigma_x$ be a k-local Hamiltonian. Then, there is a universal constant C such that*

$$\widetilde{H} = \sum_{x\in\{0,1\}^{2n}} |\lambda_x|^{\frac{2k}{k+1}} \leq C^k\|H\|.$$

Proof of Theorem 4.7. We only analyze the correctness of Theorem 9, as the complexity quickly follows from Theorem 4.6. In this proof we also use the notation of Algorithm 9. The ℓ_2-error of approximating H with $\widetilde{H}$ is

$$\|\widetilde{H} - H\|_{\ell_2}^2 = \sum_{|\lambda'_x| \le \gamma} |\lambda_x|^2 + \sum_{|\lambda'_x| \ge \gamma, |x| \le k} |\lambda_x - \lambda'_x|^2. \tag{30}$$

We show separately that the two terms are at most $O(\varepsilon^2)$. To bound the contribution of the small Pauli coefficients, we first note that by Theorem 4.6 we have that

$$|\lambda'_x| \le \gamma \quad \Longrightarrow \quad |\lambda_x| \le \gamma + \beta = O(\gamma). \tag{31}$$

Hence,

$$\sum_{|\lambda'_x| \le \gamma} |\lambda_x|^2 \le \sum_{|\lambda_x| \le O(\gamma)} |\lambda_x|^2 \le O(\gamma^{\frac{2}{k+1}}) \sum_{x \in \{0,1\}^{2n}} |\lambda_x|^{\frac{2k}{k+1}} \le \gamma^{\frac{2}{k+1}} (C^k \|H\|^2)^{\frac{2k}{k+1}} = O(\varepsilon), \tag{32}$$

where in the first inequality we used Eq. (31), in the third inequality we used Theorem 4.8 and in the last inequality that $\gamma = (\varepsilon/\|H\|^2)^{k+1} \exp(-k(k+1)/2)$. To bound the contribution of the coefficients $|\lambda_x| \ge \gamma$ we notice that there is at most $\|H\|^2/\gamma^2$ of them, because $\sum_x |\lambda_x|^2 \le \|H\|^2$. Thus,

$$\sum_{|\lambda'_x| \ge \gamma, |x| \le k} |\lambda_x - \lambda'_x|^2 \le \frac{\|H\|^2}{\gamma^2} \sup_x |\lambda_x - \lambda'_x|^2 \le \frac{\|H\|^2 \beta^2}{\gamma^2} = \varepsilon^2,$$

where in the second inequality we use the λ'_x are β-estimates of λ_x and in the last equality we use that $\beta = \gamma\varepsilon/\|H\|$.

4.3.2. Learning sparse Hamiltonians In this section we introduce our sparse Hamiltonian learner and prove its guarantees.

Algorithm 10 Sparse Hamiltonian learner

Input: Query access to the time evolution of $U(t) = e^{-itH}$, error parameter $\varepsilon \in (0, 1)$, sparsity parameter $s \in \mathbb{N}$ and failure parameter $\delta \in (0, 1)$

1: Learn $(\varepsilon/2)$-estimates λ'_x of λ_x via Algorithm 8
2: **for** $x \in \{x : \lambda'_x \ne 0\}$ **do**
3: **if** $\lambda'_x \le \varepsilon/2$ **then**
4: $\widetilde{\lambda}_x = 0$
5: **else**
6: $\widetilde{\lambda}_x = \lambda'_x$

Output: $\widetilde{H} = \sum_x \widetilde{\lambda}_x \sigma_x$

Theorem 4.9 (Sparse Hamiltonian learning). *Given an n-qubit, s-sparse Hamiltonian H, Algorithm 10 outputs another Hamiltonian $\widetilde{H} = \sum \widetilde{\lambda}_x \sigma_x$ such that, with probability $\ge 1 - \delta$, satisfies $\|H - \widetilde{H}\|_{\ell_\infty} \le \varepsilon$. The algorithm uses $\widetilde{O}(\|H\|^4/\varepsilon^4 \cdot \log(1/\delta))$ queries and $\widetilde{O}(\|H\|^2/\varepsilon^3 \cdot \log(1/\delta))$ total evolution time.*

Furthermore, if $\lambda_x = 0$, then $\widetilde{\lambda}_x = 0$. This implies that running Algorithm 10 with $\varepsilon = \varepsilon'/\sqrt{s}$ outputs $\widetilde{H}$ such that $\|H - \widetilde{H}\|_{\ell_2} \le \varepsilon'$. In this case, the algorithm uses $\widetilde{O}(\|H\|^4 s^2/\varepsilon'^4 \cdot \log(1/\delta))$ queries and $\widetilde{O}(\|H\|^2 s^{1.5}/\varepsilon'^3 \cdot \log(1/\delta))$ total evolution time.

Proof. The first part, concerning learning in the ℓ_∞ error follows from Theorem 4.6. The fact that $\lambda_x = 0$, then $\widetilde{\lambda}_x = 0$ follows from Line 3 of Algorithm 10. Finally, we note that having $\lambda_x = 0 \implies \widetilde{\lambda}_x = 0$ and $|\lambda_x - \lambda_x| \leq \varepsilon'/\sqrt{s}$, implies $\|H - \widetilde{H}\|_{\ell_2} \leq \varepsilon'$. Indeed,

$$\|H - \widetilde{H}\|_{\ell_2} = \sum_{\lambda_x \neq 0} |\lambda_x - \widetilde{\lambda}_x|^2 \leq s \sup_x |\lambda_x - \widetilde{\lambda}_x|^2 = \varepsilon'^2,$$

where in the first step we have used that $\lambda_x = 0 \implies \widetilde{\lambda}_x = 0$, in the second that $|\lambda_x - \lambda_x| \leq \varepsilon'/\sqrt{s}$ and in the third that H is s-sparse. □

We remark here that, our testing complexity is larger than our learning complexity, which might make it seem as if our testing result is trivial. But crucially, our testing algorithm here is in the much harder framework of *tolerant* testing: in this harder framework, it is unclear if proper learning algorithms imply tolerant testing algorithms.[9]

5. Testing and Learning Without Quantum Memory

5.1. Learning without memory. In this section, we use the subroutines that we established in Sect. 3.3 to remove the need for quantum memory in our learning algorithms.

Theorem 5.1. *Let H be a n-qubit traceless Hamiltonian. There are memory-less algorithms such that, with probability $\geq 1 - \delta$,*

- *they learn H up to error ε in ℓ_∞ error using $\widetilde{O}(\|H\|^8 n \log(1/\delta)/\varepsilon^8)$ queries and $\widetilde{O}(\|H\|^7 n \log(1/\delta)/\varepsilon^6)$ total evolution time,*
- *if H is k-local, they learn H in ε error in ℓ_2 norm using just $\exp(O(k^2 + k \log(\|H\|^2/\varepsilon)))\log(n/\delta)$ queries and total evolution time.*
- *if H is s-sparse, they learn H in ε error in ℓ_2 norm using just $\widetilde{O}(\|H\|^8 s^4 n \log(1/\delta)/\varepsilon^8)$ queries and $\widetilde{O}(\|H\|^7 s^3 n \log(1/\delta)/\varepsilon^6)$ total evolution time,*

Proof. The proof of the first bullet point results follow by mimicking the proofs of Algorithm 8, but substituting Pauli sampling by Lemmas 3.6 and 4.5 by Lemma 3.6. The second and third bullet points follow from the first as Theorems 4.7 and 4.9 followed from Theorem 4.6. The memory-less subroutines, Lemmas 3.6 and 3.3, incur in quadratically more queries and an extra factor due to the union bound over the set of potentially non-zero Pauli coefficients of H that appears in the proof of Lemma 3.3. Note that in the case where the Hamiltonian is promised to be local, the Pauli support is of size at most $O(n)^k$, so in this case we pay just a factor $k \log(n)$ when applying Lemma 3.3. In case that there is no promise or we are only promised sparsity, any of the 4^n Pauli can be non-zero, so we have to pay a factor n. □

5.2. Testing sparse Pauli channels via Pauli hashing. We will test the sparsity of a Hamiltonian by reducing the task to testing sparsity of a relevant Pauli channel. In this section, we thus describe how to test sparse Pauli channels via Pauli hashing, which was introduced in Sect. 3.4. Recall that a Pauli channel can be written as

$$\mathcal{E}(\rho) = \sum_{x \in \{0,1\}^{2n}} p(x) \sigma_x \rho \sigma_x,$$

[9] The seminal result of Goldreich et al. [GGR98] shows that *proper* learning algorithms implied standard property testing algorithms.

and its Pauli fidelities $\lambda(y) = \frac{1}{2^n} \operatorname{Tr}\left(\sigma_y \cdot \mathcal{E}(\sigma_y)\right)$ satisfy that $\breve{\lambda}(x) = p(x)$. We can thus test the sparsity of $\mathcal{E}$, that is it having s many non-zero error rates by testing the sparsity of the symplectic Fourier spectrum of the Pauli fidelity $\lambda : \mathbb{F}_2^{2n} \to [-1, 1]$.

5.2.1. Testing algorithm **Random coset structure.** As in Sect. 3.4.1, we define the pairwise independent hashing process corresponding to a random subgroup H of dimension t, generated by uniformly sampling $\{h_j\}_{j\in[t]}$ from $\mathbb{F}_2^{2n}$ randomly. Abusing notation, we will also call the subgroup of Weyl operators corresponding to the strings in H as H as well and similarly for the generators. We then define the buckets corresponding to $b \in \mathbb{F}_2^t$ accordingly as

$$C(b) := \{\alpha \in \mathbb{F}_2^{2n} : [\alpha, h_j] = b_j \; \forall j \in [t]\}. \tag{33}$$

Let us denote the energy of bucket $C(b)$ for each $b \in \mathbb{F}_2^t$ as the sum of the error rates in the corresponding coset $a + \mathsf{C}(H)$, expressed as

$$E(b) = \sum_{x\in a+\mathsf{C}(H)} p(x). \tag{34}$$

Energy estimation. Let the bucket $C(b)$ be the coset $a + \mathsf{C}(H)$. The energy of the bucket $E(b)$ is

$$\begin{aligned}
E(b) = \sum_{x\in a+\mathsf{C}(H)} \breve{\lambda}(x) &= \sum_{x\in a+\mathsf{C}(H)} \mathbb{E}_{z\in\mathbb{F}_2^{2n}}\left[(-1)^{[z,x]}\lambda(z)\right] \\
&= \frac{1}{4^n} \sum_{z\in\mathbb{F}_2^{2n}} \lambda(z) \sum_{x\in a+\mathsf{C}(H)} (-1)^{[z,x]} \\
&= \frac{1}{4^n} \sum_{z\in\mathbb{F}_2^{2n}} \lambda(z)(-1)^{[z,a]} \sum_{h\in\mathsf{C}(H)} (-1)^{[z,h]} \\
&= \frac{|\mathsf{C}(H)|}{4^n} \sum_{z\in\mathbb{F}_2^{2n}} \lambda(z)(-1)^{[z,a]}[z \in \mathsf{C}(\mathsf{C}(H))] \\
&= \frac{1}{|\mathsf{A}(H)|} \sum_{z\in H} \lambda(z)(-1)^{[z,a]},
\end{aligned} \tag{35}$$

where we used Fact 2.6 in the second to last equality and $\mathsf{C}(\mathsf{C}(H)) = H$ in the last equality. We now discuss how the energy of each bucket is computed efficiently using Pauli eigenstates and Pauli measurements.

Claim 5.2. *Let H be a subspace of dimension t. Then, the energy $E(b)$ of each bucket $b \in \mathbb{F}_2^t$ can be estimated upto error ε with probability $1 - \delta$ using $O(2^t/\varepsilon^2 \cdot \log(2^t/\delta))$ queries. This only requires preparation of Pauli eigenstates and Pauli measurements.*

Proof. By definition, $\lambda(z) = \frac{1}{2^n} \operatorname{Tr}(\sigma_z \mathcal{E}(\sigma_z))$ and thereby can be estimated by repeated queries involving preparation of the eigenstate of Pauli σ_z, application of the Pauli channel $\mathcal{E}$, and then measuring with respect to the Pauli σ_z. For a given z, we can thus obtain an estimate $\widetilde{\lambda}(z)$ such that $|\widetilde{\lambda}(z) - \lambda(z)| \leq \varepsilon$ with probability $1 - \delta$, using $O(1/\varepsilon^2 \log(1/\delta))$ queries as described.

For a given bucket $C(b)$, we can obtain an estimate $\widetilde{E}(b)$ of its energy $E(b)$ (Eq. (35)) by taking the empirical mean of the each of the Pauli fidelity estimates $\{\widetilde{\lambda}(z)\}_{z\in H}$

$$\widetilde{E}(b) = \frac{1}{|\mathsf{A}(H)|}\sum_{z\in H}\widetilde{\lambda}(z)(-1)^{[z,a]},$$

where $a \in \mathbb{F}_2^{2n}$ is such that the bucket $C(b)$ is equivalent to the coset $a+\mathsf{C}(H)$. Note that with an overall query complexity $O(2^t/\varepsilon^2\log(2^t/\delta))$, we can ensure that $|\widetilde{\lambda}(z)-\lambda(z)| \leq \varepsilon$ for all $z \in H$ and thereby have

$$|\widetilde{E}(b) - E(b)| \leq \frac{1}{|\mathsf{A}(H)|}\sum_{z\in H}|\widetilde{\lambda}(z)(-1)^{[z,a]} - \lambda(z)(-1)^{[z,a]}| \leq \frac{1}{|H|}\sum_{z\in H}\varepsilon = \varepsilon,$$

where we used that $|\mathsf{A}(H)| = |H|$. □

We are now in a position to state Algorithm 11 which tests s-sparsity of Pauli channels.

Algorithm 11 Testing s-sparsity of Pauli channels

Input: A Pauli channel $\mathcal{E}$, sparsity parameter s, error parameters $0 \leq 2\varepsilon_1 < \varepsilon_2 < 1$

1: Randomly sample a subgroup H of dimension $t = O(\log s)$
2: Set bucket indices $\{b_1, \ldots, b_T\} = \mathbb{F}_2^t$
3: Set $\varepsilon = (\varepsilon_2 - 2\varepsilon_1)/3$ and query budget $T = O(s/\varepsilon^2\log(s/\delta))$
4: Obtain energy estimates $\{\widetilde{E}(b)\}_{b\in\mathbb{F}_2^t}$ using Claim 5.2 upon inputs of H and budget T
5: Compute $\Gamma \leftarrow \max_{S\subseteq\mathbb{F}_2^t:|S|=s}\sum_{j\in S}\widetilde{E}(j)$

Output: If $\Gamma \geq 1 - 8/3\varepsilon_1 + 1/3\varepsilon_2$, close-to-sparse. If $\Gamma \leq 1 - 4/3\varepsilon_2 + 2/3\varepsilon_1$, output far-from-sparse.

In the next section, we will argue that the query complexity of Algorithm 11 is indeed poly(s).

5.2.2. Analysis In this section, we prove the following theorem. Our analysis follows the approach of [YZ20].

Theorem 5.3. *Let $\mathcal{E}$ be an n-qubit Pauli channel. We can test if $\mathcal{E}$ is ε_1-close to or ε_2-far from being s-sparse in diamond norm with probability ≥ 0.9 using*

$$O\left(\frac{s^2}{(\varepsilon_2-\varepsilon_1)^6}\left(\log\frac{1}{(\varepsilon_2-\varepsilon_1)} + \log s\right)\right)$$

queries to $\mathcal{E}$. These queries involve no quantum memory and only Pauli eigenstate preparations or Pauli measurements.

Our goal will be to show that Γ in Algorithm 11 is an $((\varepsilon_2 - 2\varepsilon_1)/3)$-approximation of $\mathsf{Energy}(\mathcal{E}; s)$, which is enough to determine if $\mathcal{E}$ is close to or far from being sparse thanks to Claim 2.7. Note that Γ has two sources of error. The first appears because we approximate $\mathsf{Energy}(\mathcal{E}; s)$ via $\max_{|S|=s}\sum_{j\in S}E(j)$, which we will refer to it as the *hashing error*. The hashing error will be small because thanks to Proposition 3.7 all the top $p(x)$ will be in different buckets. The second source of error comes from

approximating $E(b)$ via $\widetilde{E}(b)$. We will refer to it as *estimation error* and will be controlled thanks to Claim 5.2.

Hashing error. Let the energies of the buckets be indexed in non-increasing order by energy as $E_1 \geq E_2 \geq \cdots \geq E_{2^t}$ and label the 2^t buckets accordingly, $B(1), \ldots, B(2^t)$. Let $E'_j = \max_{x \in B(j)} p(x)$. We also assign labels $y_1, \ldots, y_{4^n}$ to $\{0,1\}^{2n}$ in a way such that the error rates satisfy as $p_{y_1} \geq p_{y_2} \geq \cdots \geq p_{y_{4^n}}$. With this notation, the hashing error is given by

$$\mathsf{err}(\mathcal{E}; H, s) = \sum_{j=1}^{s} \left(E_j - p_{y_j}\right).$$

We will prove the following upper bound to the hashing error.

Lemma 5.4. *Let $\varepsilon \in (0, 1/2]$. Suppose H is a random subgroup of dimension $t \geq \log(2s/\varepsilon^2)$, then*

$$\Pr_H \left[\mathsf{err}(\mathcal{E}; H, s) \leq 6\varepsilon\right] \geq 0.96\,.$$

To show Lemma 5.4, we first note that since $p_{y_1}, \ldots, p_{y_s}$ are the s largest error rates, it follows that

$$\mathsf{err}(\mathcal{E}; H, s) = \sum_{j=1}^{s} \left(E_j - E'_j\right) + \sum_{j=1}^{s} \left(E'_j - p_{y_j}\right) \leq \sum_{j=1}^{s} \left(E_j - E'_j\right). \tag{36}$$

Hence, to upper bound the hashing error we just have to upper bound the *collision error* $\sum_{j=1}^{s} \left(E_j - E'_j\right)$, which occurs when two or more coefficients collide in the same bucket. With that purpose, we prove the following claim regarding $\mathsf{err}_j := E_j - E'_j$.

Claim 5.5. *Suppose H is a subgroup of dimension t drawn uniformly at random. Then,*

$$\mathbb{E}_H \left[\sum_{j=1}^{s} \mathsf{err}_j\right] \leq \sqrt{\frac{2s}{2^t}}, \quad \mathsf{Var}_H \left[\sum_{j=1}^{s} \mathsf{err}_j\right] \leq \frac{2s}{2^t}.$$

Proof. We define $A_{j,i}$ as $\{0,1\}$-valued random variable for the event that $y_j \in B(i)$ or $A_{j,i} = [y_j \in B(i)]$, and $D_{j,i}$ as the indicator variable for the event that p_{y_j} is not the largest error rate in $B(i)$ or $D_{j,i} = [p_{y_j} \neq E'_i]$. We then have that

$$E_i = \sum_{j \in [2^{2n}]} p_{y_j} A_{j,i}, \quad E'_i = \sum_{j \in [2^{2n}]} p_{y_j} A_{j,i} (1 - D_{j,i}) \tag{37}$$

By the Cauchy-Schwarz inequality follows that

$$\sum_{i=1}^{s} \mathsf{err}_i \leq \sqrt{s} \sqrt{\sum_{i=1}^{s} \mathsf{err}_i^2}.$$

We can bound $\sum_{i=1}^{s} \mathsf{err}_i^2$ as follows:

$$\begin{aligned}
\sum_{i=1}^{s} \mathsf{err}_i^2 \leq \sum_{i=1}^{2^t} \left(E_i - E_i'\right)^2 &= \sum_{i=1}^{2^t} \left(\sum_{j \in [2^{2n}]} p_{y_j} A_{j,i} D_{j,i} \right)^2 \\
&= \sum_{i=1}^{2^t} \sum_{j,k \in [2^{2n}]} p_{y_j} p_{y_k} A_{j,i} A_{k,i} D_{j,i} D_{k,i} \\
&\leq \sum_{j,k \in [2^{2n}]} p_{y_j} p_{y_k} \sum_{i=1}^{2^t} A_{j,i} A_{k,i} D_{j,i} \\
&= \sum_{j,k \in [2^{2n}]} p_{y_j} p_{y_k} \sum_{i=1}^{2^t} [y_j, y_k \in B(i)][y_j \neq E'_{B(i)}] \\
&= \sum_{j,k \in [2^{2n}]} p_{y_j} p_{y_k} [y_j, y_k \in B(i) \text{ for some } i][p_{y_j} \neq E_i' \\
&\qquad \text{when } p_{y_j} \in B(i)]
\end{aligned}$$

where in the first line we used that the top s buckets is a subset of the total 2^t buckets and Eq. (37); and in the third line we used that $D_{k,i} \in \{0, 1\}$. Let $F_{j,k} := [y_j, y_k \in B(i) \text{ for some } i]$, i.e, $F_{j,k}$ indicates if y_j and y_k collide in the same bucket. Let $D_j := [p_{y_j} \neq E_i' \text{ when } p_{y_j} \in B(i)]$, i.e., D_j indicates that y_j does not correspond to the biggest error rate in its bucket. Taking expectation over H gives us

$$\begin{aligned}
\mathbb{E}_H \left[\sum_{i=1}^{s} \mathsf{err}_i^2 \right] &\leq \mathbb{E}_H \left[\sum_{j,k \in [2^{2n}]} p_{y_j} p_{y_k} F_{jk} D_j \right] \\
&= \mathbb{E}_H \left[\sum_{j \in [2^{2n}]} p_{y_j}^2 D_j \right] + \mathbb{E}_H \left[\sum_{j \neq k \in [2^{2n}]} p_{y_j} p_{y_k} F_{jk} D_j \right] \\
&\leq \underbrace{\sum_{j \in [2^{2n}]} p_{y_j}^2 \mathbb{E}_H \left[D_j \right]}_{(*)} + \underbrace{\sum_{j \neq k \in [2^{2n}]} p_{y_j} p_{y_k} \mathbb{E}_H \left[F_{jk} \right]}_{(**)}
\end{aligned} \tag{38}$$

where in the second line we used that $D_j \in \{0, 1\}$. We analyze the terms $(*)$ and $(**)$ separately. First, we deal with $(*)$,

$$\begin{aligned}
(*) = \sum_{j \in [2^{2n}]} p_{y_j}^2 \mathbb{E}_H \left[D_j \right] &= \sum_{j \in [2^{2n}]} p_{y_j}^2 \mathbb{E}_H \left[\max_{i<j} \{F_{j,i}\} \right] \leq \sum_{j \in [2^{2n}]} p_{y_j}^2 \mathbb{E}_H \left[\sum_{i<j} F_{j,i} \right] \\
\leq \sum_{j=1}^{2^{2n}} p_{y_j}^2 \frac{j-1}{2^t} \leq \frac{1}{2^t} \sum_{j=2}^{2^{2n}} \sum_{k=1}^{j-1} p_{y_j} p_{y_k} &\leq \frac{1}{2^t} \left(\sum_{j=1}^{2^{2n}} p_{y_j} \right)^2 = \frac{1}{2^t},
\end{aligned} \tag{39}$$

where the first line we have used that y_j does not correspond to the biggest error rate of its bucket if and only if there is an $i < j$ such that y_i is in the same bucket as y_j; and in the line we used Proposition 3.7 (b) and that $p_{y_j} < p_{y_k}$ if $j > k$.

The term $(**)$ can be bounded as

$$(**) = \sum_{j \neq k \in [2^{2n}]} p_{y_j} p_{y_k} \mathbb{E}_H \left[F_{jk} \right] = \frac{1}{2^t} \sum_{j \neq k \in [2^{2n}]} p_{y_j} p_{y_k} \leq \frac{1}{2^t} \left(\sum_{j \in [2^{2n}]} p_{y_j} \right)^2 = \frac{1}{2^t}, \tag{40}$$

where in the first equality we used Proposition 3.7 (b). We can now substitute Eq. (39) and Eq. (40) into Eq. (38) to obtain

$$\mathbb{E}_H \left[\sum_{i=1}^{s} \mathsf{err}_i^2 \right] \leq \frac{2}{2^t}. \tag{41}$$

In order to prove our expectation upper bound now, we use

$$\mathbb{E}_H \left[\sum_{i=1}^{s} \mathsf{err}_i \right] \leq \sqrt{s}\, \mathbb{E}_H \left[\sqrt{\sum_{i=1}^{s} \mathsf{err}_i^2} \right] \leq \sqrt{s} \sqrt{\mathbb{E}_H \left[\sum_{i=1}^{s} \mathsf{err}_i^2 \right]} \leq \sqrt{\frac{2s}{2^t}}, \tag{42}$$

where we used Jensen's inequality in the second inequality and Eq. (41) in the last inequality. This proves the result regarding expectation. To prove the result regarding variance, we note that

$$\mathrm{Var}_H \left[\sum_{i=1}^{s} \mathsf{err}_i \right] \leq \mathbb{E}_H \left[\left(\sum_{i=1}^{s} \mathsf{err}_i \right)^2 \right] \leq s \mathbb{E}_H \left[\sum_{i=1}^{s} \mathsf{err}_i^2 \right] \leq \frac{2s}{2^t}. \tag{43}$$

where we again used Eq. (41) in the last inequality. □

We are now ready to give the proof of Lemma 5.4.

Proof of Lemma 5.4. From Eq. (36), we have that $\mathsf{err}(\mathcal{E}; H, s) \leq \sum_{i=1}^{s} \mathsf{err}_i$. Then,

$$\Pr_H \left[\mathsf{err}(\mathcal{E}; H, s) \leq \varepsilon \right] \geq \Pr_H \left[\sum_{i=1}^{s} \mathsf{err}_i \leq \varepsilon \right] \tag{44}$$

Using Chebyshev's inequality along with Claim 5.5, we have that for any $a > 0$

$$\begin{aligned} & \Pr_H \left[\sum_{i=1}^{s} \mathsf{err}_i \geq \mathbb{E}_H \left[\sum_{i=1}^{s} \mathsf{err}_i \right] + a \sqrt{\mathrm{Var}_H \left[\sum_{i=1}^{s} \mathsf{err}_i \right]} \right] \leq \frac{1}{a^2} \\ & \Pr_H \left[\sum_{i=1}^{s} \mathsf{err}_i \geq \frac{\sqrt{2s} + a\sqrt{2s}}{\sqrt{2^t}} \right] \leq \frac{1}{a^2} \\ \Longrightarrow \quad & \Pr_H \left[\sum_{i=1}^{s} \mathsf{err}_i \geq \left(1 + \frac{a\sqrt{s}}{\sqrt{s}} \right) \varepsilon \right] \leq \frac{1}{a^2} \end{aligned}$$

where we substituted $t = \log(2s/\varepsilon^2)$ in the third line. Finally setting $a = 5$ and noting that $s \geq 1$, we can combine the above expression with Eq. (44) to show that

$$\Pr_H \left[\mathsf{err}(\mathcal{E}; H, s) \leq 6\varepsilon\right] \geq 0.96, \tag{45}$$

which completes the proof of the desired result. □

Estimation error. Recall that our energy estimates of the different buckets were $\{\widetilde{E}(b)\}_{b\in\mathbb{F}_2^t}$. Let the energy estimates of the buckets be indexed in non-increasing order as $\widetilde{E}_1 \geq \widetilde{E}_2 \geq \cdots \geq \widetilde{E}_{2^t}$. Let us denote the estimated energy over the top s buckets as

$$\Gamma = \sum_{j=1}^{s} \widetilde{E}_j.$$

We now bound the error between our estimate Γ and $\mathsf{Energy}(\mathcal{E}; s)$.

Claim 5.6. *Suppose H is a random subgroup of dimension t. If $t \geq \log(2s/\varepsilon^2)$, then with query complexity $O\left(\frac{s^2}{\varepsilon^6}\left(\log\frac{1}{\varepsilon} + \log s\right)\right)$, we have*

$$\Pr_H \left[|\Gamma - \mathsf{Energy}(\mathcal{E}; s)| \leq \varepsilon\right] \geq 0.92\,.$$

Proof. We note that

$$|\Gamma - \mathsf{Energy}(\mathcal{E}; s)| = \left|\sum_{j=1}^{s}(\widetilde{E}_j - p_{y_j})\right| \leq \left|\sum_{j=1}^{s}(\widetilde{E}_j - E_j)\right| + \left|\sum_{j=1}^{s}(E_j - p_{y_j})\right|$$

The second term on the right hand side is the hashing error $\mathsf{err}(\mathcal{E}; H, s)$, which by Lemma 5.4 is bounded above by $\varepsilon/2$ for $t \geq \log(2s/\varepsilon^2)$ with probability more than 0.96. The first time on the right hand side can be bounded as

$$\left|\sum_{j=1}^{s}(\widetilde{E}_j - E_j)\right| \leq \sum_{j=1}^{s}\left|\widetilde{E}_j - E_j\right| \leq \sum_{j=1}^{2^t}\left|\widetilde{E}_j - E_j\right| \leq \varepsilon/2,$$

where we used the fact that top s buckets are a subset of all the buckets in the second inequality and concluded the final inequality by using by Claim 5.2 along with a union bound for $t \geq \log(2s/\varepsilon^2)$. This consumes $O(2^{2t}/\varepsilon^2 \log(4^t)) = O\left(\frac{s^2}{\varepsilon^6}\left(\log\frac{1}{\varepsilon} + \log s\right)\right)$ queries to ensure desired error and success probability greater than 0.96. Combining the bounds on the above two sums gives us our desired result. □

Proof of main theorem on testing sparse Pauli channels. We can now complete the proof of Theorem 5.3, and thereby show correctness of Algorithm 11.

Proof of Theorem 5.3. Using Claim 5.6, we can ensure that $|\Gamma - \mathsf{Energy}(\mathcal{E}; s)| \leq (\varepsilon_2 - 2\varepsilon_1)/3$ with probability greater than 0.92, using $O\left(\frac{s^2}{(\varepsilon_2-\varepsilon_1)^6}\left(\log\frac{1}{(\varepsilon_2-\varepsilon_1)} + \log s\right)\right)$ queries. Thanks to Claim 2.7, this is enough to determine if a Pauli channels is close to or far from sparse.

5.3. Testing sparse Hamiltonians via Pauli hashing. In this section, we describe how to test the sparsity of Hamiltonians by testing the sparsity of a related Pauli channel and then using Pauli hashing. Given a Hamiltonian $H = \sum \lambda_x \sigma_x$ its time evolution channel at time t is given by

$$\mathcal{H}_t(\rho) = U(t)\rho U(t)^\dagger, \tag{46}$$

where $U(t) = e^{-itH}$. Considering the Pauli decomposition of the unitary $U(t) = \sum_{x\in\{0,1\}^{2n}} \widehat{U}(x)\sigma_x$ and its complex conjugate $U(t) = \sum_{x\in\{0,1\}^{2n}} \overline{\widehat{U}(x)}\sigma_x$, we can write

$$\mathcal{H}_t(\rho) = \sum_{x,y\in\{0,1\}^{2n}} \widehat{U}(x)\overline{\widehat{U}(y)}\sigma_x\rho\sigma_y = \sum_{x,y\in\{0,1\}^{2n}} \kappa_{xy}\sigma_x\rho\sigma_y, \tag{47}$$

where κ_{xy} are coefficients corresponding to the Pauli characters $\sigma_x\rho\sigma_y$. Because of Taylor theorem (Eq. (1)), the diagonal coefficients κ_{xx} are given by

$$\kappa_{xx} = \begin{cases} |\widehat{U}(x)|^2 = \lambda_x^2 t^2 + o(t^3) & , x \neq 0, \\ 1 - \sum_{x\in\{0,1\}^{2n}\setminus 0^{2n}} |\widehat{U}(x)|^2 = 1 - t^2 \sum_{x\in\{0,1\}^{2n}\setminus 0^{2n}} \lambda_x^2 + o(t^3) & , x = 0, \end{cases} \tag{48}$$

We can however convert the Hamiltonian evolution channel $\mathcal{H}_t$ into a diagonal Pauli channel by applying *Pauli twirling* [CB19, BW23] as follows

$$\mathcal{H}_t^{\mathcal{T}}(\rho) = \mathbb{E}_x[\sigma_x \mathcal{H}_t(\sigma_x\rho\sigma_x)\sigma_x]. \tag{49}$$

Particularly, $\mathcal{H}_t^{\mathcal{T}}$ takes the following form

$$\mathcal{H}_t^{\mathcal{T}}(\rho) = \sum_{x\in\{0,1\}^{2n}} p_t^{\mathcal{T}}(x)\sigma_x\rho\sigma_x, \tag{50}$$

where we have denoted the error rates of the channel $\mathcal{H}_t^{\mathcal{T}}$ as $\{p_t^{\mathcal{T}}(x)\}$ and they are related to κ_{xx} as $p(x) = \kappa_{xx}$.

We now provide intuition for how testing the sparsity of n-qubit Hamiltonians can be accomplished by testing the sparsity of the Pauli channel $\mathcal{H}_t^{\mathcal{T}}$. Let us order the Pauli coefficients of $U(t)$ excluding $\widehat{U}(0^{2n})$ in a non-increasing order as $|\widehat{U}_1|^2 \geq |\widehat{U}_2|^2 \geq \cdots \geq |\widehat{U}_{2^{2n}-1}|^2$. From Lemma 3.2, we noted that the following quantity was a good proxy for testing whether H was close to being s-sparse or not

$$\mathsf{TopEnergy}(t; s) := |\widehat{U}(0^{2n})|^2 + \sum_{j=1}^{s} |\widehat{U}_j|^2, \tag{51}$$

Moreover, we observe that [10]

$$\mathsf{TopEnergy}(t; s) = p_t^{\mathcal{T}}(0^{2n}) + \mathsf{Energy}(\mathcal{H}_t^{\mathcal{T}}; s), \tag{52}$$

where $\mathsf{Energy}(\mathcal{E}; s)$ is estimated in Algorithm 11 as part of testing Pauli channels $\mathcal{E}$ and is equal to the sum of the top s error rates of the channel. We can thus test the sparsity of n-qubit Hamiltonians by testing the sparsity of the Pauli channel $\mathcal{H}_t^{\mathcal{T}}$, which leads to the following result which we will prove shortly.

[10] This is true provided that 0^{2n} is not among the top s error rates of $\mathcal{H}_t^{\mathcal{T}}$. We will make a more general statement later that removes this assumption.

Theorem 5.7. *Let H be an n-qubit Hamiltonian such that $\|H\|_{\text{op}} \leq 1$. We can test if H is ε_1-close to (in normalized Frobenius norm) or ε_2-far from s-sparse Hamiltonians with probability > 0.9 using*

$$O\left(\frac{s^{14}}{(\varepsilon_2^2-\varepsilon_1^2)^{18}}\left(\log\frac{s^2}{(\varepsilon_2^2-\varepsilon_1^2)^3}+\log s\right)\right)$$

queries to the evolution operator $U(t)=\exp(-iHt)$ for a choice of $t=O((\varepsilon_2^2-\varepsilon_1^2)/s)$ and with a total evolution time of $O((s^{13}\log s)/(\varepsilon_2^2-\varepsilon_1^2)^{17})$.

5.3.1. Testing algorithm We now proceed as we had for testing sparse Pauli channels in the earlier section. We define the relevant random coset structure and how to estimate energy across different buckets, which will be the main component of our testing algorithm. The main difference here is that we do not have direct access to the relevant Pauli channel i.e., the Pauli-twirled Hamiltonian evolution channel $\mathcal{H}_t^{\mathcal{T}}$ but we show that we can still compute energy estimates across the different buckets corresponding to $\mathcal{H}_t^{\mathcal{T}}$ by querying the Hamiltonian evolution channel $\mathcal{H}_t$ itself.

Hashing and coset structure. As in Sect. 3.4.1, we define the pairwise independent hashing process corresponding to a random subgroup G of dimension d, generated by uniformly sampling $\{g_j\}_{j\in d}$ from $\mathbb{F}_2^{2n}$ randomly. In a slight abuse of notation, we will also call the subgroup of Weyl operators corresponding to the strings in G as G as well and similarly for the generators. We then define the buckets corresponding to $b\in\mathbb{F}_2^d$ accordingly as

$$C(b):=\{x\in\mathbb{F}_2^{2n}:[x,g_j]=b_j\,\forall j\in[d]\}. \tag{53}$$

Let us denote the energy of bucket $C(b)$ for each $b\in\mathbb{F}_2^d$ as the sum of the error rates of the Pauli channel $\mathcal{H}_t^{\mathcal{T}}$ (Eq. (50)) obtained from Pauli twirling of the Hamiltonian evolution channel in the corresponding coset $a+\mathsf{C}(G)$, expressed as

$$E(b)=\sum_{x\in a+\mathsf{C}(G)} p_t^{\mathcal{T}}(x). \tag{54}$$

Energy estimation. Recall from Eq. (35) that the enery of each bucket can be expressed as

$$E(b)=\frac{1}{|\mathsf{A}(G)|}\sum_{z\in G}\lambda_t^{\mathcal{T}}(z)(-1)^{[z,a]}, \tag{55}$$

where $\lambda_t^{\mathcal{T}}$ are the Pauli fidelities of the channel $\mathcal{H}_t^{\mathcal{T}}$. We now discuss how the energy of each bucket is computed efficiently using Pauli eigenstates and Pauli measurements.

Claim 5.8. *Let G be a random subspace of dimension d. Then using Algorithm 12, the energy $E(b)$ of each bucket $b\in\mathbb{F}_2^d$ can be estimated to within error ε with probability $1-\delta$ with $O(2^d/\varepsilon^2\cdot\log(2^d/\delta))$ queries. This only requires preparation of Pauli eigenstates and Pauli measurements.*

Proof. Let the eigenvalue measurement outcome obtained in line 10 of Algorithm 12 be denoted as $(-1)^w$ with $w \in \{0,1\}$. The joint probability of obtaining $a \in \mathbb{F}_2^{2n}$ (which are sampled uniformly at random) and obtaining measurement outcome $w \in \{0,1\}$ is then

$$\Pr(a,w) = \frac{1}{2^{2n}} \operatorname{Tr}\left[\frac{\mathrm{Id} + (-1)^w \sigma_z}{2} \sigma_a \mathcal{H}_t\left(\sigma_a \frac{\mathrm{Id}_{2^n} + \sigma_z}{2^n} \sigma_a\right) \sigma_a\right]$$

Marginalizing over $a \in \mathbb{F}_2^{2n}$ gives us the probability of obtaining measurement outcome w

$$\begin{aligned}
\Pr(w) &= \mathbb{E}_{a \in \mathbb{F}_2^{2n}} \operatorname{Tr}\left[\frac{\mathrm{Id} + (-1)^w \sigma_z}{2} \sigma_a \mathcal{H}_t\left(\sigma_a \frac{\mathrm{Id}_{2^n} + \sigma_z}{2^n} \sigma_a\right) \sigma_a\right] \\
&= \operatorname{Tr}\left[\frac{\mathrm{Id} + (-1)^w \sigma_z}{2} \mathbb{E}_{a \in \mathbb{F}_2^{2n}}\left[\sigma_a \mathcal{H}_t\left(\sigma_a \frac{\mathrm{Id}_{2^n} + \sigma_z}{2^n} \sigma_a\right) \sigma_a\right]\right] \\
&= \operatorname{Tr}\left[\frac{\mathrm{Id} + (-1)^w \sigma_z}{2} \mathcal{H}_t^{\mathcal{T}}\left(\frac{\mathrm{Id}_{2^n} + \sigma_z}{2^n}\right)\right] \\
&= \frac{1}{2} \operatorname{Tr}\left[\mathcal{H}_t^{\mathcal{T}}\left(\frac{\mathrm{Id}_{2^n} + \sigma_z}{2^n}\right)\right] + \frac{(-1)^w}{2} \operatorname{Tr}\left[\sigma_z \mathcal{H}_t^{\mathcal{T}}\left(\frac{\mathrm{Id}_{2^n} + \sigma_z}{2^n}\right)\right] \\
&= \frac{1}{2} + \operatorname{Tr}\left[\mathcal{H}_t^{\mathcal{T}}\left(\frac{\sigma_z}{2^n}\right)\right] + \frac{(-1)^w}{2} \operatorname{Tr}\left[\sigma_z\right] + \frac{(-1)^w}{2} \operatorname{Tr}\left[\sigma_z \mathcal{H}_t^{\mathcal{T}}\left(\frac{\sigma_z}{2^n}\right)\right] \\
&= \frac{1 + (-1)^w \lambda_t^{\mathcal{T}}(z)}{2}
\end{aligned}$$

where the third equality follows from the definition of Pauli twirling and definition of $\mathcal{H}_t^{\mathcal{T}}$ in Eq. (50), and we use the definition of Pauli fidelity in the last equality. Observing that $\mathbb{E}[(-1)^w] = \lambda_t^{\mathcal{T}}(z)$, we can then obtain an estimate $\lambda_t^{\mathcal{T}}(z)$ using the outcomes of w and taking an empirical mean of $(-1)^w$. The estimate will be within ε error with probability $1-\delta$ using $O(1/\varepsilon^2 \log(1/\delta))$ queries. We can now conclude as we had done in the proof of Claim 5.2 and noting Eq. (55). □

We are now ready to give our tester, which is showcased in Algorithm 13.

5.3.2. Analysis In this section, we prove Theorem 13 and thereby show correctness of Algorithm 13. **Hashing error.** As in Sect. 5.2.2, we first analyze the error in estimating the energy of the top s error rates $\mathsf{Energy}(\mathcal{H}_t^{\mathcal{T}}; s)$ via our random hashing process. We assume here that all the energy estimates across the cosets are exact and analyze the estimation error later.

Let the energies of the buckets $\{C(b)\}_{b \in \mathbb{F}_2^d \setminus 0^d}$ be indexed in non-increasing order by energy as $E_1 \geq E_2 \geq \cdots \geq E_{2^d-1}$. We pay special attention to the bucket corresponding to the coset $C(0^d)$ which will always include the error rate of $p(0^{2n})$ due to the construction of our buckets. We will also use $\tilde{E}_j = \max_{x \in C(b)} p(x)$ to denote the energy of largest coefficient hashed into the jth bucket denoted by $C(b)$ for some $b \in \mathbb{F}_2^d$. The true values of the energies are clearly the error rates $\{p_t^{\mathcal{T}}(x)\}_{x \in \mathbb{F}_2^{2n}}$ themselves. We will also order the error rates $\{p_t^{\mathcal{T}}(x)\}_{\in \mathbb{F}_2^{2n} \setminus 0^{2n}}$(i.e., excluding $p(0^{2n})$) in a non-increasing order as $p_1^{\mathcal{T}} \geq p_2^{\mathcal{T}} \geq \cdots \geq p_{2^{2n}-1}^{\mathcal{T}}$. The goal is to obtain an estimate of $\mathsf{TopEnergy}(t; s)$ (Eq. (52)) from the constructed buckets. In particular, we want to

Algorithm 12 Energy estimation of buckets for Hamiltonian evolution

Input: Budget $T = O(2^d/\varepsilon^2 \log(2^d/\delta))$, access to unitary evolution $U(t) = \exp(-iHt)$, evolution time t, subspace G of dimension $d = O(\log s)$, set of buckets $B = \{b\} \subseteq \mathbb{F}_2^d$

1: Initialize energy estimates $\widetilde{E}(b) = 0$
2: Initialize fidelity estimates $\widetilde{\lambda}_t^{\mathcal{T}}(z) = 0$ and counter $m(z)$ for each $z \in G$
3: **for** $z \in G$ **do**
4: **for** query $t = 1, \ldots, T$ **do**
5: Uniformly sample at random $a \in \mathbb{F}_2^{2n}$
6: Prepare Pauli eigenstate ρ_z of σ_z
7: Apply Pauli σ_a to ρ_z
8: Apply unitary evolution $U(t) = \exp(-iHt)$
9: Apply Pauli σ_a
10: Measure current state with respect to the Pauli basis σ_z to obtain eigenvalue $\gamma \in \{\pm 1\}$
11: Update $\widetilde{\lambda}_t^{\mathcal{T}}(z) \leftarrow \widetilde{\lambda}_t^{\mathcal{T}}(z) + \gamma$, $m(z) \leftarrow m(z) + 1$
12: Set $\widetilde{\lambda}_t^{\mathcal{T}}(z) \leftarrow \frac{1}{m(z)} \widetilde{\lambda}_t^{\mathcal{T}}(z)$
13: Set $\widetilde{E}(b) \leftarrow \frac{1}{|G|} \sum_{z \in H} \widetilde{\lambda}_t^{\mathcal{T}}(z)(-1)^{[z,a]}$ where $C(b)$ corresponds to coset $a + \mathsf{C}(G)$, for each $b \in B$

Output: $\{\widetilde{E}(b)\}_{b \in B}$

Algorithm 13 Testing s-sparsity of Hamiltonians

Input: Access to unitary Hamiltonian evolution $U(t) = \exp(-iHt)$, evolution time t, sparsity s, error parameters $0 < \varepsilon_1 < \varepsilon_2 \leq 1$

1: Randomly sample a subgroup H of dimension $d = O(\log s)$
2: Set bucket indices $\{b\} = \mathbb{F}_2^d$
3: Set $\varepsilon = (\varepsilon_2^2 - \varepsilon_1^2)^3/6s^2$ and query budget $T = O(\mathrm{poly}(s)/\varepsilon^2 \log(1/\varepsilon))$
4: Obtain energy estimates $\{\widetilde{E}(b)\}_{b \in \mathbb{F}_2^d}$ using Algorithm 12 upon inputs of $U(t)$, evolution time t, and budget T
5: Compute $\Gamma \leftarrow \widetilde{E}(0^d) + \max_{S \subset \mathbb{F}_2^d \setminus 0^d : |S| = s} \sum_{j \in S} \widetilde{E}(j)$

Output: If $\Gamma \geq 1 - \varepsilon_1^2 \frac{(\varepsilon_2^2 - \varepsilon_1^2)^2}{s^2} - \frac{1}{2} \frac{(\varepsilon_2^2 - \varepsilon_1^2)^3}{s^2}$, output close-to-sparse. If $\Gamma \leq 1 - \varepsilon_2^2 \frac{(\varepsilon_2^2 - \varepsilon_1^2)^2}{s^2} + \frac{1}{2} \frac{(\varepsilon_2^2 - \varepsilon_1^2)^3}{s^2}$, output far-from-sparse.

determine the value of $p(0^{2n}) + \sum_{j=1}^{s} p_{y_j}$ to obtain such an estimate. The *hashing error* is then accordingly defined as

$$\mathsf{err}(\mathcal{H}_t^{\mathcal{T}}; G, s) = \left(E(0^d) - p(0^{2n})\right) + \sum_{j=1}^{s} \left(E_j - p_{y_j}\right),$$

where we denoted the error by $\mathsf{err}(\cdot)$ along with noting the random subspace G involved in Pauli hashing. As $p_1^{\mathcal{T}}, \ldots, p_s^{\mathcal{T}}$ are the s largest error rates in $\mathbb{F}_2^{2n} \setminus 0^{2n}$, it follows that

$$\begin{aligned}\mathsf{err}(\mathcal{H}_t^{\mathcal{T}}; G, s) &= \left(E(0^d) - p(0^{2n})\right) + \sum_{j=1}^{s} \left(E_j - E'_j\right) + \sum_{j=1}^{s} \left(E'_j - p_{y_j}\right) \\ &\leq \left(E(0^d) - \tilde{E}(0^d)\right) + \sum_{j=1}^{s} \left(E_j - E'_j\right)\end{aligned}$$

The following corollary of Lemma 5.4 is then immediate.

Corollary 5.9. *Fix $\varepsilon \in (0, 1/2]$. If G is a random subgroup of dimension $t \geq \log(2s/\varepsilon^2)$, then*

$$\Pr_G \left[\mathsf{err}(\mathcal{H}_t^{\mathcal{T}}; G, s) \leq 6\varepsilon \right] \geq 0.96.$$

We remark that the proof of Corollary 5.9 is very similar to that of Lemma 5.4 and is thus not included. Note that the proof of Lemma 5.4 bounds the error of any s buckets. In particular, it could be used over the $s+1$ buckets that include error rates corresponding to 0^{2n} (or bucket of $\mathsf{C}(G)$) and the top s buckets different from $\mathsf{C}(G)$.

Estimation error. Recall that our energy estimates of the different buckets were $\{\widetilde{E}(b)\}_{b\in\mathbb{F}_2^t}$. Let the energies of the buckets $\{C(b)\}_{b\in\mathbb{F}_2^d\setminus 0^d}$ be indexed in non-increasing order as $\widetilde{E}_1 \geq \widetilde{E}_2 \geq \cdots \geq \widetilde{E}_{2^d-1}$. Let us denote the estimated energy over the bucket $C(0^{2n})$ and the top s buckets (excluding $C(0^{2n})$) as

$$\Gamma = \widetilde{E}(0^d) + \max_{S\subset\mathbb{F}_2^d\setminus 0^d : |S|=s} \sum_{j\in S} \widetilde{E}(j) = \widetilde{E}(0^d) + \sum_{j=1}^{s} \widetilde{E}_j,$$

where the last equality follows from the definition of $\widetilde{E}_j$ above. We now bound the error between our estimate Γ and $\mathsf{TopEnergy}_H(t)$ (Eq. (51)) by immediately applying Claim 5.6 to obtain the following corollary.

Corollary 5.10. *Suppose G is a random subgroup of dimension d. If $d \geq \log(2s/\varepsilon^2)$, then with query complexity $O\left(s^2/\varepsilon^6 \left(\log \frac{1}{\varepsilon} + \log s\right)\right)$, we have*

$$\Pr_H \left[|\Gamma - \mathsf{TopEnergy}(t; s)| \leq \varepsilon \right] \geq 0.92\,.$$

Proof of main theorem on testing sparse Hamiltonians. We can now complete the proof of Theorem 5.7, and thereby show correctness of Algorithm 13.

Proof of Theorem 5.7. Let $t = O((\varepsilon_2^2 - \varepsilon_1^2)/s)$. By Lemma 3.2, we have that if H is ε_1-close to being sparse, then

$$\mathsf{TopEnergy}(t; s) \geq 1 - \varepsilon_1^2 \frac{(\varepsilon_2^2 - \varepsilon_1^2)^2}{s^2} - \frac{1}{3}\frac{(\varepsilon_2^2 - \varepsilon_1^2)^3}{s^2},$$

while if H is ε_2-far from s-sparse, then

$$\mathsf{TopEnergy}(t; s) \leq 1 - \varepsilon_2^2 \frac{(\varepsilon_2^2 - \varepsilon_1^2)^2}{s^2} + \frac{1}{3}\frac{(\varepsilon_2^2 - \varepsilon_1^2)^3}{s^2}.$$

For our tester, it then suffices to estimate $\mathsf{TopEnergy}_H(t)$ up to error

$$\varepsilon = \frac{(\varepsilon_2^2 - \varepsilon_1^2)^3}{6s^2}.$$

Using Corollary 5.10 and for the specified t, we can ensure that $|\Gamma - \mathsf{TopEnergy}_H(t)| \leq \varepsilon$ with probability greater than 0.92, using $O\left(\frac{s^{14}}{(\varepsilon_2^2-\varepsilon_1^2)^{18}}\left(\log \frac{s^2}{(\varepsilon_2^2-\varepsilon_1^2)^3} + \log s\right)\right)$ queries. The decision rules of Γ in Algorithm 13 for accepting/rejecting are then evident.

6. Lower Bounds for Learning

Below we prove lower bounds on learning sparse and local Hamiltonians given query access to the time-evolution operator. In comparison to prior work of [BCO24b, Theorem 5.1] the class of Hamiltonians witnessing our lower bound also applies to algorithms which are diagonal and encode just Boolean functions; in contrast the hard instance in [BCO24b] applies to Hamiltonians involve application of Haar random unitaries. In order to prove our lower bounds, we use the following generic lemma, which will allow us to reduce the problem of proving lower bounds for Hamiltonian testing/learning problems to a question about Boolean function analysis. Below we will be talking of the standard oracle model in query complexity, i.e.,

$$O_f : |x, 0\rangle \to |x, f(x)\rangle.$$

Theorem 6.1 ([GAW19, Theorem 14]). *Let $p : X \to [0, 1]$, and suppose $U_p : \mathcal{H} \otimes \mathcal{H}_{aux.} \to \mathcal{H} \otimes \mathcal{H}_{aux.}$ is a probability oracle with an n-qubit auxiliary Hilbert space $\mathcal{H}_{aux.} = \mathbb{C}^{2^n}$. Let $\varepsilon \in (0, 1/3)$, then we can implement an ε-approximate phase oracle O such that for any phase oracle O_p defined as*

$$O_p : |0\rangle \mapsto \sum_x \sqrt{p(x)}|x\rangle,$$

and for all $|\psi\rangle \in \mathcal{H}$

$$\| O|\psi\rangle|0\rangle^{\otimes(n+a)} - \mathrm{O}_p|\psi\rangle|0\rangle^{\otimes(n+a)} \|_{\mathrm{op}} \leq \varepsilon,$$

using $O(\log(1/\varepsilon))$ applications of U_p and $U_p^\dagger$, with $a = O(\log\log(1/\varepsilon))$, where

$$U_p : |x, 0\rangle \to |x\rangle \otimes \left(\sqrt{p(x)}|\psi_g(x)\rangle|0\rangle + \sqrt{1 - p(x)}|\psi_b(x)\rangle|1\rangle\right),$$

where $|\psi_g(x)\rangle, |\psi_b(x)\rangle$ are arbitrary orthogonal states.

Lemma 6.2. *Let $t \geq 0$. Let $f : \{0, 1\}^n \to \{-1, 1\}$ be a Boolean function. Given quantum query access to an oracle O_f, then we can simulate*

$$H_f = e^{it \sum_{S \in \{0,1\}^n} \widehat{f}(S)|S\rangle\langle S|}$$

for time $t \in \mathbb{R}$ with precision ε making $\widetilde{O}(t \log(1/\varepsilon))$ queries to U_f.

Proof. We first observe that, using the phase kickback trick, with access to O_f, one can also implement the phase oracle. Applying O_f on a uniform superposition, we get $\frac{1}{\sqrt{2^n}} \sum_x |x, f(x)\rangle$. Next, one can obtain $\frac{1}{\sqrt{2^n}} \sum_x f(x)|x\rangle$ with probability $1/2$: replace $f(x) \in \{-1, 1\}$ by $(1 - f(x))/2 \in \{0, 1\}$ unitarily, apply the Hadamard transform to the last qubit and measure it. With probability $1/2$ we obtain the outcome 0, in which case our procedure rejects. Otherwise the remaining state is $\frac{1}{\sqrt{2^n}} \sum_x f(x)|x\rangle$. So from here onwards, we can assume that we have access to

$$O_f : |0\rangle \to \frac{1}{\sqrt{2^n}} \sum_x f(x)|x\rangle.$$

Below we show that one use the oracle U to construct a probability oracle U_p:

$$
\begin{aligned}
U_f : |x\rangle|0\rangle|0\rangle \quad &\overset{Had}{\mapsto} \quad |x\rangle|0\rangle|+\rangle \\
&\overset{O_f}{\mapsto} \quad |x\rangle \frac{1}{\sqrt{2^n}} \sum_{y\in X} f(y)|y\rangle|+\rangle \\
&\overset{Had^n}{\mapsto} \quad |x\rangle \sum_{y\in X} \widehat{f}(y)|y\rangle|+\rangle \\
&\overset{H_f}{\mapsto} \quad |x\rangle \left(\widehat{f}(x)|x\rangle|-\rangle + \sum_{y\neq x} \widehat{f}(y)|y\rangle|+\rangle \right) \\
&\overset{Had}{\mapsto} \quad |x\rangle \left(\widehat{f}(x)|x\rangle|1\rangle + \sum_{y\neq x} \widehat{f}(y)|y\rangle|0\rangle \right) \\
&\overset{Swap}{\mapsto} \quad |x\rangle \left(\widehat{f}(x)|x\rangle|1\rangle + \sum_{y\neq x} \widehat{f}(y)|y\rangle|0\rangle \right).
\end{aligned}
$$

As Theorem 6.1 shows we can simulate a fractional phase query O_f^r where $r := t/\lceil |t| \rceil$ with precision $\varepsilon/\lceil |t| \rceil$ making $O(\log(t/\varepsilon))$ queries to U_f. Observe that $\lceil |t| \rceil$ consecutive applications of O_f^r give O_f^t, which is exactly the Hamiltonian simulation unitary that we wanted to implement. □

The same proof as the lemma above implies the following corollary as well.

Corollary 6.3. *Let $t \geq 0$. Let $p : \{0,1\}^n \to [0,1]$ be a distribution. Given quantum query access to an oracle O_p we can simulate a Hamiltonian corresponding to the probability distribution*

$$H_p = e^{it \sum_{S\in\{0,1\}^n} p(S)|S\rangle\langle S|}$$

for time $t \in \mathbb{R}$ with precision ε making $\widetilde{O}(t \log(1/\varepsilon))$ queries to U_p.

6.1. Adaptive coherent memoryless learning lower bound. To prove our main theorem, we use the following facts, starting with a well known bound on the size of an ε-net of the n-dimensional sphere.

Fact 6.4 *([Tao12, Exercise 2.3.1]) For every $d \geq 1$ and any $0 < \varepsilon < 1/2$ there exists an ε-net of the sphere S^{d-1} of cardinality at least $t = (c/\varepsilon)^d$, i.e., there exists $\{v_1, \ldots, v_t\} \subseteq S^{d-1}$ such that $\|v_i - v_j\|_2 \geq \varepsilon$ for all $i \neq j$.*

Theorem 6.5. *There exists a class of Hamiltonians with $\|H\|_2 \leq 1$ such that learning s-sparse n-qubit Hamiltonians without quantum memory upto error ε using time step t, need to make $\Omega\big(s(\log 1/\varepsilon)/(t \log s)\big)$ adaptive quantum queries.*

Proof. Let's assume that the Hamiltonian is supported on the *first* $(\log s)$ qubits and only has support on the σ_0, σ_3 (i.e., it is diagonal), so the total sparsity equals $2^{(\log s)} = s$ (note that in the original learning algorithm, the support is *unknown* to the learner so

we are proving a lower bound on a simpler problem here). Let the corresponding Pauli coefficients be $\{\lambda_x : x \in \{0,3\}^{(\log s)}\}$. Now consider an ε-net on these qubits to be the set of vectors $\{h^1, \ldots, h^t\} \subseteq \{0,3\}^{\log s}$ where $t = (1/\varepsilon)^s$. Now, let us consider the class of Hamiltonians $\mathcal{H}$, given by

$$H_i = \sum_{x \in \{0,3\}^{(\log s)}} h^i(x)\sigma_x.$$

Observe that the unitary evolution corresponding to this Hamiltonian is given by

$$U_p = e^{it \sum_{x \in \{0,1\}^n} p(x)|x\rangle\langle x|}$$

By Corollary 6.3, every learning algorithms that made queries to U_p at time t, can be converted (with a factor t-overhead) to an algorithm that only is given *quantum queries to* p, i.e., given access to the standard oracle model $O_p : |0\rangle \to \sum_x \sqrt{p(x)}|x\rangle$. Clearly a learning algorithm for learning H_i by making queries to U_p upto error-ε in the ℓ_2-distance implies that the learning algorithm can *identify* i since the coefficient vectors h^i form an ε-net.

Now we use Holevo's bound to conclude the proof (we will use a version that appeared in Nayak's work [Nay99] and also recently in a work of Chen et al. [CNY23]). Since each quantum query contains $(\log s)$-qubits of information (since we assumed that the Hamiltonian was identity on the remaining $n - \log s$ qubits), if there is a k-query algorithm that *identifies* i as above, then

$$k \cdot (\log s) \geq \log |\mathcal{H}|,$$

and now using the lower bound from Fact 6.4, that implies a lower bound of

$$k \geq s/(\log s) \cdot \log(1/\varepsilon).$$

Overall, this implies an $\Omega(s/(t \log s) \cdot \log(1/\varepsilon))$ lower bound on the learning problem. □

One inherent weakness in the proof technique above is, the $(\log s)$-factor in the denominator cannot be removed since Holevo's bound is a generic statement about transmitting arbitrary quantum states. In the next section, we are able to use a more careful analysis and show that one can get rid of the dependence on $(\log s)$-albeit in a weaker model of learning.

6.2. Non-adaptive incoherent memoryless learning lower bound.

Theorem 6.6. *Learning s-sparse n-qubit Hamiltonians without quantum memory upto error $1/2s$ using time step t, requires*

$$\Omega\big((s \log s)/t\big)$$

non-adaptive quantum queries.

In order to prove this theorem, we will need the following

Proof. In order to prove this theorem, we first construct our hard instance of Hamiltonians based on Boolean functions. To this end, we embed Boolean functions as Hamiltonians in a natural way: for every $f : \{0,1\}^n \to \{-1,1\}$, let

$$H_f = \sum_{S \subseteq [n]} \widehat{f}(S) \prod_{i \in S} Z_i,$$

in which case $H_f|x\rangle = f(x)|x\rangle$ for every basis state $x \in \{0,1\}^n$. Such an embedding was considered and shown to be useful in [Had21]. Observe that the unitary evolution corresponding to this Hamiltonian is given by

$$U_f = e^{it \sum_{S \in \{0,1\}^n} \widehat{f}(S)|S\rangle\langle S|}$$

By Lemma 6.2, every learning algorithms that made queries to U_f at time t, can be converted to an algorithm that only is given *quantum queries to* f, i.e., given access to the standard oracle model $O_f : |x,0\rangle \to |x, f(x)\rangle$.[11].

So from here onwards, we prove a lower bound on the number of quantum queries required for our learning task, and that complexity divided by t will be our eventual Hamiltonian learning lower bound. Our next step is to construct our hard set of Boolean functions. To this end, for every $(\log s)$-dimensional subspace V, let $f_V(x) = [x \in V^\perp]$, in which case it is well-known that

$$\widehat{f}(S) = [x \in V]/s,$$

see [O'D14] for a proof. In particular, note that the number of T for which $\widehat{f}(T) \neq 0$ equals s. In particular, the corresponding Hamiltonian H_f whose Pauli coefficients are precisely $\widehat{f}(T)$, has sparsity s.

Our third step now is to consider the class of Boolean functions

$$\mathcal{C} = \{f_V : \{0,1\}^n \to \{0,1\} | f_V(x) = [x \in V^\perp] \text{ s.t.} V \text{ is a } (\log s) - \text{dimensional subspace}\}.$$

We next show that every non-adaptive quantum learning algorithm for learning the unknown f (given quantum query access O_f), needs to make $\Omega(k)$ queries. The proof of this is similar to the information-theoretic proof in [ADW18]. We prove the lower bound for $\mathcal{C}$ using a three-step information-theoretic technique. Let $\mathbf{A}$ be a random variable that is uniformly distributed over $\mathcal{C}$. Suppose $\mathbf{A} = f_V$, and let $\mathbf{B} = \mathbf{B}_1 \ldots \mathbf{B}_T$ be T quantum queries

$$|\psi^i_{f_V}\rangle = \sum_{x \in \{0,1\}^n} \sqrt{\alpha_i(x)}|x, f_V(x)\rangle,$$

for $f_V \in \mathcal{C}$, where the amplitudes could potentially depend on the ith query, but independent of $i-1$ different measurement outcomes. The random variable $\mathbf{B}$ is a function of the random variable $\mathbf{A}$. The following upper and lower bounds on $I(\mathbf{A} : \mathbf{B})$ are similar to [ADW18, Theorem 12] and we omit the details of the first two steps here.

1. $I(\mathbf{A} : \mathbf{B}) \geq \Omega(\log |\mathcal{C}|)$ because $\mathbf{B}$ allows one to recover $\mathbf{A}$ with high probability.
2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$ using a chain rule for mutual information.

[11] Technically, we need the phase oracle, but the bit-oracle and phase oracle are equivalent up to a constant overhead in query complexity when allowed controlled operations

3. $I(\mathbf{A}:\mathbf{B}_1) \leq O(n \cdot \eta_a)$.

Proof (of 3). Since $\mathbf{AB}$ is a classical-quantum state, we have

$$I(\mathbf{A}:\mathbf{B}_1) = S(\mathbf{A}) + S(\mathbf{B}_1) - S(\mathbf{AB}_1) = S(\mathbf{B}_1),$$

where the first equality is by definition and the second equality uses $S(\mathbf{A}) = \log|\mathcal{C}|$ since $\mathbf{A}$ is uniformly distributed over $\mathcal{C}$, and $S(\mathbf{AB}_1) = \log|\mathcal{C}|$ since the matrix

$$\sigma = \frac{1}{|\mathcal{C}|}\sum_{f_V \in \mathcal{C}} |c\rangle\langle c| \otimes |\psi_{f_V}\rangle\langle\psi_{f_V}|$$

is block-diagonal with $|\mathcal{C}|$ rank-1 blocks on the diagonal. It thus suffices to bound the entropy of the (vector of singular values of the) reduced state of $\mathbf{B}_1$, which is

$$\rho = \frac{1}{|\mathcal{C}|}\sum_{f_V \in \mathcal{C}} |\psi_{f_V}\rangle\langle\psi_{f_V}|.$$

For notational convenience, let $\eta_{\mathsf{a}} = \mathbb{E}_{c,c'\in\mathcal{C}} \Pr_x[c(x) \neq c'(x)]$. Also, let $\sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_{2^{n+1}-1} \geq 0$ be the singular values of ρ. Since ρ is a density matrix, these form a probability distribution. Now observe that $\sigma_0 \geq 1 - \eta_a$: consider the vector $u = \frac{1}{|\mathcal{C}|}\sum_{c'\in\mathcal{C}}|\psi_{c'}\rangle$ and observe that

$$\begin{aligned}
u^\top \rho u &= \frac{1}{|\mathcal{C}|^3}\sum_{V,V',V''\in\mathcal{C}} \langle\psi_{f_V}|\psi_{f_{V'}}\rangle\langle\psi_{f_V}|\psi_{f_{V''}}\rangle \\
&= \mathbb{E}_V\Big[\mathbb{E}_{V'}[\langle\psi_{f_V}|\psi_{f_{V'}}\rangle]\Big]\cdot\Big[\mathbb{E}_{V''}[\langle\psi_{f_V}|\psi_{f_{V''}}\rangle]\Big] \\
&\geq \Big(\mathop{\mathbb{E}}_{V,V'}[\langle\psi_{f_V}|\psi_{f_{V'}}\rangle]\Big)\cdot\Big(\mathop{\mathbb{E}}_{V,V''}[\langle\psi_{f_V}|\psi_{f_{V''}}\rangle]\Big) \\
&= \Big(\mathop{\mathbb{E}}_{f_V,f_{V'}\in\mathcal{C}} \Pr_x[f_V(x) = f_{V'}(x)]\Big)^2 \geq 1 - 2\eta_a,
\end{aligned}$$

where the first inequality is by Chebyshev's sum inequality (since all the inner products are non-negative) and the second inequality follows from the definition of η_a. Hence we have that $\sigma_0 = \max_u\{u^\top\rho u/u^\top u\} \geq 1 - 2\eta_a$ (where we used that $\|u\|_2 \leq 1$).

Let $\mathbf{N} \in \{0, 1, \ldots, 2^{n+1}-1\}$ be a random variable with probabilities $\sigma_0, \sigma_1, \ldots, \sigma_{2^{n+1}-1}$, and $\mathbf{Z}$ an indicator for the event "$\mathbf{N} \neq 0$." Note that $\mathbf{Z} = 0$ with probability $\sigma_0 \geq 1 - 2\eta_a$, and $H(\mathbf{N} \mid \mathbf{Z} = 0) = 0$. By a similar argument as in [ADW18, Theorem 15], we have

$$\begin{aligned}
S(\rho) &= H(\mathbf{N}) = H(\mathbf{N},\mathbf{Z}) = H(\mathbf{Z}) + H(\mathbf{N} \mid \mathbf{Z}) \\
&= H(\sigma_0) + \sigma_0 \cdot H(\mathbf{N} \mid \mathbf{Z} = 0) + (1-\sigma_0)\cdot H(\mathbf{N} \mid \mathbf{Z} = 1) \\
&\leq H(\eta_a) + \eta_a(n+1) \\
&\leq O(\eta_a(n + \log(1/\eta_a))
\end{aligned}$$

using $H(\alpha) \leq O(\alpha\log(1/\alpha))$.

Combining these three steps implies $T = \Omega(\log|\mathcal{C}|/(n\eta_a))$.

It now remains to bound $|\mathcal{C}|, \eta_a$. To this end, we prove bounds on both quantities below.

Claim 6.7. *The number of distinct* $(\log s)$*-dimensional subspaces of* $\mathbb{F}_2^n$ *is at least* $s^{\Omega(n-\log s)}$.

Proof. For simplicity below, let $d = \log s$. We can specify a d-dimensional subspace by giving d linearly independent vectors in it. The number of distinct sequences of d linearly independent vectors is exactly $(2^n-1)(2^n-2)(2^n-4)\cdots(2^n-2^{d-1})$, because once we have the first t linearly independent vectors, with span $\mathcal{S}_t$, then there are $2^n - 2^t$ vectors that do not lie in $\mathcal{S}_t$.

However, we are double-counting certain subspaces in the argument above, since there will be multiple sequences of vectors yielding the same subspace. The number of sequences yielding a fixed d-dimensional subspace can be counted in a similar manner as above and we get $(2^d-1)(2^d-2)(2^d-4)\cdots(2^d-2^{d-1})$. So the total number of subspaces is

$$\frac{(2^n-1)(2^n-2)\cdots(2^n-2^{d-1})}{(2^d-1)(2^d-2)\cdots(2^d-2^{d-1})} \geq \frac{(2^n-2^{d-1})^d}{(2^d-1)^d} \geq 2^{\Omega((n-d)d)} = s^{\Omega(n-\log s)},$$

where we used $d = \log s$ □

Next, it remains to *upper* bound η_a. To this end, first observe that

$$\begin{aligned}\eta_{\mathsf{a}} \leq \max_{c,c' \in \mathcal{C}} \Pr_x[c(x) \neq c'(x)] &= \max_{V,V'} \Pr_x[c_V(x) + c_{V'}(x) \neq 0] \\ &= \max_{V} \Pr_x[c_V(x) = 1] = \frac{1}{2^n} \cdot |V^{\perp}| = 1/s.\end{aligned}$$

Putting everything together, we have shown that, *exact* learning the concept class $\mathcal{C}$ requires

$$T = \Omega\left(\frac{\log|\mathcal{C}|}{n\eta_a}\right) \geq \Omega\left(\frac{(\log s)\cdot(n-\log s)}{n\cdot 1/s}\right) = \Omega(s\log s).$$

In order to conclude the proof of the theorem, note that every algorithm that satisfies[12]

$$\begin{aligned}\varepsilon \geq \sum_x |\lambda_x - \lambda'_x|^2 &= \sum_S (\widehat{f}(S) - \widehat{f'}(S))^2 \\ &= \mathbb{E}_x[|f(x) - f'(x)|^2] = \frac{1}{2^n}\sum_x [f(x) \neq f'(x)] = d(f, f')\end{aligned}$$

We finally conclude by using [HR16, Claim 2.2]

$$d(f_V, f_{V'}) = \Pr_x[f_V(x) \neq f_{V'}(x)] \geq 1/(2s),$$

so learning this function to error $< 1/(2s)$ implies exact learnability. □

[12] Note that technically the learning algorithm, need not output a Boolean function $f' : \{0,1\}^n \to \{-1,1\}$, but in our setting since it *knows* that the unknown Hamiltonian is characterized by a Boolean function, even if the algorithm outputs a real-valued function, then rounding it to a bit would also be a good approximation.

Acknowledgement S.A. and A.D. thank the Institute for Pure and Applied Mathematics (IPAM) for its hospitality throughout the long program "Mathematical and Computational Challenges in Quantum Computing" in Fall 2023 during which part of this work was initiated. This work was done in part while S.A. was visiting the Simons Institute for the Theory of Computing, supported by DOE QSA grant #FP00010905. This research was supported by the Europea union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 945045, and by the NWO Gravitation project NETWORKS under grant no. 024.002.003. We thank Amira Abbas, Francesco Anna Mele, Andreas Bluhm, Jop Briët, Matthias Caro, Nunzia Cerrato, Aadil Oufkir, and Daniel Liang for useful comments and discussions. A.D. thanks Patrick Rall for multiple conversations on stabilizer subgroups and Pauli twirling. A.D. thanks Isaac Chuang for discussions on the problem of testing Hamiltonians. F.E.G. is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2047/1—390685813. We thank the referees for their detailed comments.

References

[AAKS21] Anshu, A., Arunachalam, S., Kuwahara, T., Soleimanifar, M.: Sample-efficient learning of interacting quantum systems. Nat. Phys. **17**(8), 931–935 (2021). https://doi.org/10.1038/s41567-021-01232-0

[ACQ22] Aharonov, D., Cotler, J., Qi, X.-L.: Quantum algorithmic measurement. Nat. Commun. **13**(1), 1–9 (2022). https://doi.org/10.1038/s41467-021-27922-0

[ADW18] Arunachalam, S., De Wolf, R.: Optimal quantum sample complexity of learning algorithms. J. Mach. Learn. Res. **19**(71), 1–36 (2018)

[AG04] Aaronson, S., Gottesman, D.: Improved simulation of stabilizer circuits. Phys. Rev. A **70**(5), 052328 (2004). https://doi.org/10.1103/PhysRevA.70.052328

[BBRV02] Bandyopadhyay, S., Boykin, P.O., Roychowdhury, V., Vatan, F.: A new proof for the existence of mutually unbiased bases. Algorithmica **34**, 512–528 (2002). https://doi.org/10.1007/s00453-002-0980-7

[BCO24a] Bluhm, A., Caro, M.C., Oufkir, A.: Hamiltonian property testing (version 1) (2024). arXiv preprint arXiv:2403.02968v1

[BCO24b] Bluhm, A., Caro, M.C., Oufkir, A.: Hamiltonian property testing (version 2) (2024). arXiv preprint arXiv:2403.02968v2

[Bla09] Blais, E.: Testing juntas nearly optimally. In: Proceedings of the 41st Annual ACM symposium on Theory of Computing (STOC), pp. 151–158. ACM (2009). https://doi.org/10.1145/1536414.1536437

[BLMT23] Bakshi, A., Liu, A., Moitra, A., Tang, E.: Learning quantum hamiltonians at any temperature in polynomial time (2023). arXiv:2310.02243

[BLMT24] Bakshi, A., Liu, A., Moitra, A., Tang, E.: Structure learning of hamiltonians from real-time evolution. In: FOCS'24 (2024)

[Can20] Canonne, C.L.: A short note on learning discrete distributions (2020). arXiv preprint arXiv:2002.11457

[Car23] Caro, M.C.: Learning quantum processes and hamiltonians via the Pauli transfer matrix (2023). arXiv:2212.04471

[CB19] Cai, Z., Benjamin, S.C.: Constructing smaller Pauli twirling sets for arbitrary error channels. Sci. Rep. **9**(1), 11281 (2019)

[CCHL22] Chen, S., Cotler, J., Huang, H.-Y., Li, J.: Exponential separations between learning with and without quantum memory. In: 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 574–585 IEEE (2022). https://doi.org/10.1109/FOCS52979.2021.00063

[CG23] Chen, S., Gong, W.: Efficient Pauli channel estimation with logarithmic quantum memory 9 (2023). arXiv:2309.14326

[CGY24] Chen, S., Gong, W., Ye, Q.: Optimal tradeoffs for estimating Pauli observables (2024). arXiv preprint arXiv:2404.19105

[CLL24] Chen, S., Li, J., Liu, A.: An optimal tradeoff between entanglement and copy complexity for state tomography. In: Proceedings of the 56th Annual ACM Symposium on Theory of Computing, pp. 1331–1342 (2024)

[CNY23] Chen, T., Nadimpalli, S., Yuen, H.: Testing and learning quantum juntas nearly optimally. In: Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1163–1185. SIAM (2023)

[CW23] Castaneda, J., Wiebe, N.: Hamiltonian learning via shadow tomography of pseudo-Choi states (2023). arXiv:2308.13020

[CZSJ22] Chen, S., Zhou, S., Seif, A., Jiang, L.: Quantum advantages for Pauli channel estimation. Phys. Rev. A **105**, 032435 (2022). https://doi.org/10.1103/PhysRevA.105.032435

[dSLCP11] da Silva, M.P., Landon-Cardinal, O., Poulin, D.: Practical characterization of quantum devices without tomography. Phys. Rev. Lett. **107**(21), 210404 (2011). https://doi.org/10.1103/PhysRevLett.107.210404

[DOS23] Dutkiewicz, A., O'Brien, T.E., Schuster, T.: The advantage of quantum control in many-body Hamiltonian learning (2023). arXiv:2304.07172

[DPW+23] Dutt, A., Pednault, E., Wu, C.W., Sheldon, S., Smolin, J., Bishop, L., Chuang, I.L.: Active learning of quantum system Hamiltonians yields query advantage. Phys. Rev. Res. **5**, 033060 (2023). https://doi.org/10.1103/PhysRevResearch.5.033060

[EIS22] Eskenazis, A., Ivanisvili, P., Streck, L.: Low-degree learning and the metric entropy of polynomials (2022). arXiv preprint arXiv:2203.09659

[FOS23] Fawzi, O., Oufkir, A., França, D.S.: Lower bounds on learning Pauli channels (2023). arXiv:2301.09192

[FW20] Flammia, S.T., Wallman, J.J.: Efficient estimation of Pauli channels. ACM Trans. Quantum Comput. (2020). https://doi.org/10.1145/3408039

[GAW19] Gilyén, A., Arunachalam, S., Wiebe, N.: Optimizing quantum optimization algorithms via faster quantum gradient computation. In: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1425–1444. SIAM (2019)

[GCC24] Gu, A., Cincio, L., Coles, P.J.: Practical Hamiltonian learning with unitary dynamics and Gibbs states. Nat. Commun. (2024). https://doi.org/10.1038/s41467-023-44008-1

[GFWC12] Granade, C.E., Ferrie, C., Wiebe, N., Cory, D.G.: Robust online Hamiltonian learning. New J. Phys. **14**(10), 103013 (2012). https://doi.org/10.1088/1367-2630/14/10/103013

[GGR98] Goldreich, O., Goldwasser, S., Ron, D.: Property testing and its connection to learning and approximation. JACM **45**(4), 653–750 (1998)

[GOS+11] Gopalan, P., O'Donnell, R., Servedio, R.A., Shpilka, A., Wimmer, K.: Testing Fourier dimensionality and sparsity. SIAM J. Comput. **40**(4), 1075–1100 (2011). https://doi.org/10.1137/100785429

[Got98] Gottesman, D.: The Heisenberg representation of quantum computers. arXiv preprint arXiv:quant-ph/9807006 (1998)

[Gut24] Gutiérrez, F.E.: Simple algorithms to test and learn local hamiltonians (2024). arXiv preprint arXiv:2404.06282

[Had21] Hadfield, S.: On the representation of boolean and real functions as Hamiltonians for quantum computing. ACM Trans. Quantum Comput. **2**(4), 1–21 (2021)

[HBCP15] Holzäpfel, M., Baumgratz, T., Cramer, M., Plenio, M.B.: Scalable reconstruction of unitary processes and hamiltonians. Phys. Rev. A **91**(4), 042129 (2015)

[HCP23] Huang, H.-Y., Chen, S., Preskill, J.: Learning to predict arbitrary quantum processes. PRX Quantum **4**(4), 040337 (2023). https://doi.org/10.1103/PRXQuantum.4.040337

[Hig08] Higham, N.J.: Functions of Matrices: Theory and Computation. SIAM (2008)

[HKP20] Huang, H.-Y., Kueng, R., Preskill, J.: Predicting many properties of a quantum system from very few measurements. Nat. Phys. **16**(10), 1050–1057 (2020)

[HKT22] Haah, J., Kothari, R., Tang, E.: Optimal learning of quantum Hamiltonians from high-temperature Gibbs states. In: 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS), pp. 135–146. IEEE (2022). https://doi.org/10.1109/FOCS54457.2022.00020

[HMG+25] Hu, H.-Y., Ma, M., Gong, W., Ye, Q., Tong, Y., Flammia, S.T., Yelin, S.F.: Ansatz-free Hamiltonian learning with Heisenberg-limited scaling (2025). arXiv preprint arXiv:2502.11900

[HR16] Haviv, I., Regev, O.: The list-decoding size of Fourier-sparse Boolean functions. ACM Trans. Computat. Theory (TOCT) **8**(3), 1–14 (2016)

[HTFS23] Hsin-Yuan Huang, Yu., Tong, D.F., Yuan, S.: Learning many-body Hamiltonians with Heisenberg-limited scaling. Phys. Rev. Lett. **130**(20), 200403 (2023). https://doi.org/10.1103/PhysRevLett.130.200403

[IBF+20] Innocenti, L., Banchi, L., Ferraro, A., Bose, S., Paternostro, M.: Supervised learning of time-independent Hamiltonians for gate design. New J. Phys. **22**(6), 065001 (2020)
[LTN+23] Li, H., Tong, Y., Ni, H., Gefen, T., Ying, L.: Heisenberg-limited Hamiltonian learning for interacting bosons (2023). arXiv:2307.04690
[LW22] LaBorde, M.L., Wilde, M.M.: Quantum algorithms for testing Hamiltonian symmetry. Phys. Rev. Lett. **129**(16), 160503 (2022)
[MBC+23] Möbus, T., Bluhm, A., Caro, M.C., Werner, A.H., Rouzé, C.: Dissipation-enabled bosonic Hamiltonian learning via new information-propagation bounds (2023). arXiv:2307.15026
[MFPT24] Ma, M., Flammia, S.T., Preskill, J., Tong, Y.: Learning k-body Hamiltonians via compressed sensing (2024). arXiv preprint arXiv:2410.18928
[MGE12] Magesan, E., Gambetta, J.M., Emerson, J.: Characterizing quantum gates via randomized benchmarking. Phys. Rev. A **85**, 042311 (2012). https://doi.org/10.1103/PhysRevA.85.042311
[Nay99] Nayak, A.: Optimal lower bounds for quantum automata and random access codes. In: 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039), pp. 369–376. IEEE (1999)
[NS12] Negahban, S., Shah, D.: Learning sparse boolean polynomials. In: 2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 2032–2036. IEEE (2012)
[O'D14] O'Donnell, R.: Analysis of Boolean Functions. Cambridge University Press, Cambridge (2014)
[ORSFW23] Onorati, E., Rouzé, C., Stilck França, D., Watson, J.D.: Efficient learning of ground and thermal states within phases of matter (2023). arXiv:2301.12946
[Pre18] Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum **2**, 79 (2018)
[PRW22] Pallavoor, R.K.S., Raskhodnikova, S., Waingarten, E.: Approximating the distance to monotonicity of boolean functions. Random Struct. Algorithms **60**(2), 233–260 (2022)
[RSF23] Rouzé, C., Stilck França, D.: Learning quantum many-body systems from a few copies (2023). arXiv:2107.03333
[SFMD+24] Stilck França, D., Markovich, L.A., Dobrovitski, V.V., Werner, A.H., Borregaard, J.: Efficient and robust estimation of many-qubit Hamiltonians. Nat. Commun. **15**, 311 (2024). https://doi.org/10.1038/s41467-023-44012-5
[SMCG16] Sheldon, S., Magesan, E., Chow, J.M., Gambetta, J.M.: Procedure for systematically tuning up cross-talk in the cross-resonance gate. Phys. Rev. A **93**, 060302(R) (2016). https://doi.org/10.1103/PhysRevA.93.060302
[SY23] She, A., Yuen, H., Bounds, U.P.T.L., by Polynomials. In: 14th Innovations in Theoretical Computer Science Conference (ITCS: volume 251 of Leibniz International Proceedings in Informatics (LIPIcs), pp. 96:1–96:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2023). https://doi.org/10.4230/LIPIcs.ITCS.2023.96
[Tao12] Tao, T.: Topics in Random Matrix Theory, vol. 132. American Mathematical Society, Providence (2012)
[BW23] van den Berg, E., Wocjan, P.: Techniques for learning sparse Pauli-Lindblad noise models (2023). arXiv preprint arXiv:2311.15408
[VZ23] Volberg, A., Zhang, H.: Noncommutative Bohnenblust-Hille inequalities. Math. Ann. (2023). https://doi.org/10.1007/s00208-023-02680-0
[Wan11] Wang, G.: Property testing of unitary operators. Phys. Rev. A **84**, 052328 (2011). https://doi.org/10.1103/PhysRevA.84.052328
[WKR+22] Wilde, F., Kshetrimayum, A., Roth, I., Hangleiter, D., Sweke, R., Eisert, J.: Scalably learning quantum many-body Hamiltonians from dynamical data (2022). arXiv:2209.14328
[YSHY23] Yu, W., Sun, J., Han, Z., Yuan, X.: Robust and efficient Hamiltonian learning. Quantum **7**, 1045 (2023). https://doi.org/10.22331/q-2023-06-29-1045
[Yu21] Yu, N.: Sample efficient identity testing and independence testing of quantum states. In: 12th Innovations in Theoretical Computer Science Conference (ITCS,: volume 185 of Leibniz International Proceedings in Informatics (LIPIcs), pp. 11:1–11:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.ITCS.2021.11
[YZ20] Yaroslavtsev, G., Zhou, S.: Fast Fourier sparsity testing. In: Symposium on Simplicity in Algorithms, pp. 57–68. SIAM (2020)
[Zha24] Zhao, A.: Learning the structure of any Hamiltonian from minimal assumptions (2024). arXiv preprint arXiv:2410.21635
[ZYLB21] Zubida, A., Yitzhaki, E., Lindner, N.H., Bairey, E.: Optimal short-time measurements for Hamiltonian learning (2021). arXiv:2108.08824

Communicated by A. Childs