Hamiltonian Complexity in the Thermodynamic Limit*

Dorit Aharonov
dorit.aharonov@gmail.com
The Hebrew University
Jerusalem, Israel

Sandy Irani
irani@ics.uci.edu
University of California, Irvine
USA

ABSTRACT

Despite immense progress in quantum Hamiltonian complexity in the past decade, little is known about the computational complexity of quantum physics at the thermodynamic limit. In fact, even defining the problem properly is not straight forward. We study the complexity of estimating the ground energy of a fixed, translationally-invariant (TI) Hamiltonian in the thermodynamic limit, to within a given precision; this precision (given by n the number of bits of the approximation) is the sole input to the problem. Understanding the complexity of this problem captures how difficult it is for a physicist to measure or compute another digit in the approximation of a physical quantity in the thermodynamic limit. We show that this problem is contained in $\text{FEXP}^{\text{QMA-EXP}}$ and is hard for $\text{FEXP}^{\text{NEXP}}$. This means that the problem is *doubly* exponentially hard in the size of the input.

As an ingredient in our construction, we study the problem of computing the ground energy of translationally invariant finite 1D chains. A single Hamiltonian term, which is a fixed parameter of the problem, is applied to every pair of particles in a finite chain. In the finite case, the length of the chain is the sole input to the problem and the task is to compute an approximation of the ground energy. No thresholds are provided as in the standard formulation of the local Hamiltonian problem. We show that this problem is contained in $\text{FP}^{\text{QMA-EXP}}$ and is hard for FP^{NEXP}. Our techniques employ a circular clock structure in which the ground energy is calibrated by the length of the cycle. This requires more precise expressions for the ground states of the resulting matrices than was required for previous QMA-completeness constructions and even exact analytical bounds for the infinite case which we derive using techniques from spectral graph theory. To our knowledge, this is the first use of the circuit-to-Hamiltonian construction which shows hardness for a function class.

CCS CONCEPTS

• **Theory of computation → Quantum complexity theory**.

KEYWORDS

Hamiltonian Complexity, Thermodynamic Limit

*We recommend that the reader consult the full version of this paper for a complete description of the construction and proofs. The full version is available at: https://arxiv.org/abs/2107.06201 [2]

STOC '22, June 20–24, 2022, Rome, Italy

ACM ISBN 978-1-4503-9264-8/22/06.
https://doi.org/10.1145/3519935.3520067

ACM Reference Format:
Dorit Aharonov and Sandy Irani. 2022. Hamiltonian Complexity in the Thermodynamic Limit. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC '22), June 20–24, 2022, Rome, Italy.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3519935.3520067

1 INTRODUCTION

Kitaev's fundamental QMA-completeness result [16] initiated the field of quantum Hamiltonian complexity [12]; however its setting is still very far from the problems that naturally arise in condensed matter physics. Subsequent work brought QMA-completeness results closer to physical settings by extending from the general geometry of Kitaev's local Hamiltonians to 2D [18], and even 1D [1] lattices; Gottesman and Irani [15] showed that the 1D result holds even for translationally invariant (TI) systems, and even when the only input is N, the size of the system.

Despite this important and fundamental progress, such complexity results are still far from capturing the primary challenges in computational many-body physics. Physicists study finite systems by necessity, but the problem which is typically of greatest interest is the estimation of physical quantities (energy density, two body correlations, etc.) in the *Thermodynamic limit* (TL)[1]. Their focus is not the estimation of physical quantities as a *function* of the size of the system, as in the QMA completeness results, but instead, they study a *particular* quantity of a Hamiltonian whose local terms are *fixed*, as $N \longmapsto \infty$.

The breakthrough undecidability result for computing the spectral gap by Cubitt, Perez-Garcia, Wolf [10] and its follow up [6] provided the first study of the *computability* of Hamiltonian problems in the TL. To the best of our knowledge, the only existing result about *computational complexity* in the TL, is that of [15] who study this as a side result to their main finite case result. More specifically, the TL problem they study is parameterized by three polynomials, r, p and q. The input is an integer N in binary representation and a Hamiltonian term h acting on a pair of d-dimensional particles whose entries are integer multiples of $1/r(N)$. They show that it is a QMA-EXP-complete problem to determine whether the ground energy density of the Hamiltonian resulting from applying h to every pair of particles in an infinite 1D chain is below $1/p(N)$ or above $1/p(N) + 1/q(N)$. However, importantly, [15] (as well as [6, 10]) study the TL when the Hamiltonian term applied to each pair of particles is *input dependent*. While the input-dependent Hamiltonian settings makes sense in the context of studying gappedness as a function of the Hamiltonian parameters (as in [6, 10]; see also the studies of *phase diagrams* for gappedness [7, 8]), it seems much less justifiable in the common physical scenario of approximating quantities of the ground state in the TL. In this context, physicists

[1] This question was highlighted by I. Cirac in an online discussion in a SIMONS institute quantum workshop, 2017

usually treat say, the AKLT model, as a different problem than, say, the Ising model.

Here we initiate the study of computational complexity in the TL for fixed, input-independent physical systems. An immediate problem arises: it is not clear how to even define the problem, namely, how to associate computational complexity to a problem whose quantity of interest is merely a single, fixed number. Our natural approach is to have the input specify only the desired *precision* to which the number – here the ground energy density in the TL – is computed, in terms of number of bits. The required precision is the only input to the problem. In order to show computational hardness, we have to show how to encode a full language into that single number. Note that this challenge does not arise in the case of input-dependent terms addressed in [15], where the reduction can encode the answer to a decision problem on input x into the input-dependent ground energy of the Hamiltonian resulting from the TI term h_x.

1.1 Problem Definition and Results

We will formally define the problem for the 2D grid; there are natural extensions to 1D and higher dimensions. The problem in 2D is parameterized by the dimension of each individual particle d, and two $d^2 \times d^2$ Hermetian matrices h^{row} and h^{col} denoting the energy interaction between two neighboring particles in the horizontal and vertical directions on a 2D grid. Then $H_{2D}(N)$ is the Hamiltonian of an $N \times N$ 2D grid of d-dimensional particles, where the same fixed $d^2 \times d^2$ local terms, h^{row} and h^{col}, are applied to every pair in the horizontal and vertical directions. $\lambda_0(H_{2D}(N))$ is the ground energy of $H_{2D}(N)$. The ground energy density of the system in the TL is defined as the following limit (we prove this limit always exists in the full version [2]):

Definition 1.1. **Energy density in the TL:** *We define α_0 as the limit*

$$\alpha_0 = \lim_{N\to\infty} \frac{\lambda_0(H_{2D}(N))}{N^2}. \quad (1)$$

The main problem we consider is to compute α_0 to within a given precision specified by the input:

Definition 1.2. Function-GED-2D for (h^{row}, h^{col})
Input: *An integer n expressed in binary.*
Output: *A number α such that $|\alpha - \alpha_0| \leq 1/2^n$.*

At first sight, it might seem counter intuitive that this problem is hard as the ground energy density for a fixed Hamiltonian in the TL is just a single number. A hardness result would need to embed a hard computational problem for all instances into a *specific number* α_0. We address this by exploiting the fact that the ground energy density in the TL is an infinite precision number, and we can use different portions of its binary representation to encode the solution to different instances of the problem from which we are reducing. Our main result is:

Theorem 1.3. Function-GED-2D *is hard for $FEXP^{NEXP}$ under Karp reductions, and is contained in $FEXP^{QMA\text{-}EXP}$.*

A remark is due regarding our chosen definition of the problem Function-GED-2D. While it would have been possible to consider a decision version of this problem, such as some variant of determining the n^{th} bit of α_0, we believe the function version more naturally describes the problem encountered in physics. Moreover, computing the n^{th} bit essentially requires computing bits 1 through $n-1$, as one can measure the n^{th} bit of the energy for a particular state, but in order to verify that the state being measured is close enough to the ground state, it seems necessary to verify that the first n bits of the energy for the given state correspond to the true ground energy density. We believe the computational complexity of determining a particular bit of α_0 would still be characterized by an oracle class, which constitutes the hardest challenge in our proof.

As in [6, 10], the proof is based on Kitaev's circuit-to-Hamiltonian construction, and works by embedding finite 1D chains into the 2D infinite lattice using Robinson tiles. The problem for the infinite grid thus reduces to a problem for finite 1D chains. However, the constructions in [6, 10] require a "two-threshold version" of the finite 1D problem; by this we mean the standard QMA-type problem, in which one needs to decide whether a quantity is larger than some threshold or smaller than another. Therefore [6, 10] can directly apply techniques from the finite case (the main result) of [15], which addresses this two-threshold setting in the 1D finite TI case. However, the hardness result given here (Theorem 1.3) requires a different type of finite 1D problem, where the task is to approximate the ground energy *to some given precision*, and no threshold is given.

To this end we define an approximation version of the 1D finite problem and characterize its complexity. Our results in the finite case pertain to 1D systems, but they can be naturally generalized for higher dimensions. In the 1D case, there is a single $d^2 \times d^2$ Hermitian matrix h which parameterizes the problem, and $H_{1D}(N)$ is the Hamiltonian resulting from applying h to every pair of neighboring particles in a $1D$ chain of length N. The function version of the finite TI Hamiltonian problem in 1D (Function-TIH-1D) is defined as follows:

Definition 1.4. Function-TIH-1D for h and constant c
Input: *An integer N expressed in binary.*
Output: *A number λ such that $|\lambda - \lambda_0(H_{1D}(N))| \leq 1/N^c$.*

The following theorem encapsulates our results for Function-TIH:

Theorem 1.5. *Function-TIH-1D problem is hard for FP^{NEXP} under Karp reductions, and is contained in $FP^{QMA\text{-}EXP}$.*

The proof of Theorem 1.5, which comprises the main technical effort of this paper, requires strengthening the finite TI results of [15] to handle approximation problems rather than two-threshold type problems. We note that [5] improves on [15] by reducing the dimension of the particles significantly. However, critically, their Hamiltonian is not input-independent. Ambainis [4], and later also [13, 14], studied a related class of problems of approximating quantities of groundstates, called APX-SIM, which they argue are better physically motivated than the standard two-threshold type local Hamiltonian problems. In those approximation problems, as in the one studied here, the absence of a given threshold presents an inherent challenge: one needs to verify that the state being measured is close enough to the ground state. As a result, the upper bounds on the complexity of these problems all require some form

of binary search with queries to a QMA or QMA-EXP oracle [4]. Indeed the natural complexity classes for such problems are *oracular* ones. This oracular structure poses a technical challenge for lower bounds (assuming the lower bounds are proven in the stronger setting of Karp reductions), due to the fact that "no" answers from the oracle cannot be verified. To overcome this challenge, we make use of a technique pioneered by Krentel [17, 19] who proved that the optimization version of certain NP-hard problems are complete for $\mathrm{FP}^{\mathrm{NP}}$. Our main technical contribution is implementing a TI version of Krentel's technique with sufficiently precise ground energy estimations as needed for the infinite case. We elaborate on how we do this in Subsection 1.2.

We remark regarding the exponential difference in complexity between the finite and the infinite case (Theorems 1.5 and 1.3). Roughly, the ground energy density in the TL in 2D can be estimated to within $\pm 1/N$ by solving a a finite grid of size $O(N^2)$ by $O(N^2)$. This fact is a by-product of the proof that the limit in (1) always exists, which is given in the full version [2]. Since the input n to the FUNCTION-GED-2D problem requires precision $1/2^n$ and is specified using $\log n$ bits, the complexity is doubly exponential in the input size. In the finite case, the size of the system itself (N) is given in binary, so the complexity is only singly exponential. From an expressibility perspective, in the finite case, every value of N can be used to encode the solution to an instance of the problem from which we are reducing. By contrast, in the infinite case, the most efficient reduction we can hope for is where each bit of α_0 encodes an answer to a computational problem. In this case, the system size has to double for each input encoded.

1.2 Proof Overview: The Finite Case

We start by giving an overview of the proof of Theorem 1.5 for computing the ground energy of finite 1D TI Hamiltonians. More details and references to the lemmas in the paper are given in Subsections 2.4- 2.8. The finite construction is used in the infinite case (Theorem 1.3) as 1D finite Hamiltonians are layered on top of a Robinson tiling of the infinite grid. An overview of the infinite case is given in Subsection 1.3.

We now consider an arbitrary $f \in \mathrm{FP}^{\mathrm{NEXP}}$, and describe how the reduction to Function-TIH-1D works. First, f is associated with a *fixed* Hamiltonian term h that operates on two d-dimensional particles. Let H_N denote the Hamiltonian on a chain of N d-dimensional particles resulting from applying h to each neighboring pair in the chain. The reduction maps an input string x of length n for the function f, to a positive integer $N = N(x)$ such that $N(x)$ can be computed in time polynomial in n. We will show that there is a polynomial q and a polynomial time classical algorithm, which for any x can compute $f(x)$ given a $1/q(n)$-approximation of $\lambda_0(H_N)$ (namely a value E, where $|E - \lambda_0(H_N)| \leq 1/q(n)$).

In the circuit-to-Hamiltonian construction, the Hilbert space of the entire chain consists of a Hilbert space which encodes the state of the computation, tensored with a Hilbert space that contains a clock which regulates the process of transitioning from one configuration to another. In the absence of any additional penalty terms, the ground energy is 0 and is achieved by the state that is a uniform superposition of all states in the computation, entangled with the clock state for that point in time: $\sum_{t=0}^{L} |t\rangle|\phi_t(\text{init})\rangle$. The $|t\rangle$ denotes the state of the clock and $|\phi_t(\text{init})\rangle$ denotes the state of the computation after starting in state $|\text{init}\rangle$ and progressing for t clock steps. In our construction. the ground state of the Hamiltonian encodes the history of a computation which simulates the polynomial time Turing Machine that computes the function f, with access to a NEXP oracle.

As mentioned above, the oracle calls pose a challenge in the circuit-to-Hamiltonian construction: the *no* guesses of the oracle responses cannot be verified. To overcome this, we use Krentel's accounting scheme [17, 19] that applies a cost to every string y representing guesses for the sequence of responses to all the oracle queries made. The accounting scheme needs to ensure that the minimum cost y is equal to the correct sequence of oracle responses, $\tilde{y}$. *yes* and *no* guesses are treated differently, due to the fact that the verifier can check *yes* instances (and thus incorrect *yes* guesses can incur a very high cost), but *no* guesses, cannot be directly verified. In Krentel's scheme, *no* guesses incur a more modest cost, whether correct or not, and their cost must decrease exponentially. This is because the oracle queries are adaptive; an incorrect oracle response could potentially change all the oracle queries made in the future and so it is important that the penalty for an incorrect guess on the i^{th} query is higher than the energy that could potentially be saved on all future queries. Thus, the costs range from a constant to exponential in m, where m is the number of oracle calls; our main challenge is that a fixed translationally-invariant Hamiltonian cannot directly encode these costs in penalty terms.

We address this issue as follows. The computation consists of repeated loops, each of which lasts $p(N)$ steps. In each repetition the computation simulates the verifier for all the *yes* oracle guesses and imposes an energy cost if any of these computations rejects (i.e., the *yes* guess was wrong). We define a function $T(x, y)$ (discussed below) and enforce that the number of loop repetitions is $2T(x, y) + 1$, making the total length of the computation $L = (2T(x, y) + 1) \cdot p(N)$ steps. The clock is circular, making eigenvalues analysis easier, and hence we refer to the entire computation as a *cycle*. The Hamiltonian is thus block diagonal in those cycles, where each cycle, or block, corresponds to a computation initiated with different input parameters: the guess string y, a guess T (in unary) of $T(x, y)$, a string w corresponding to the witnesses needed for verifying the *yes* guesses of the oracle, and finally the initial configuration for the computation, which we denote by v-init. The ground energy of the Hamiltonian is the minimum over the ground values of these blocks.

To achieve the large penalty for incorrect *yes* guesses, we introduce a penalty for each verification computation that ends with REJECT. This results in a periodic cost occurring once per iteration. We use spectral graph theory to obtain a close-to-tight lower bound on the lowest eigenvalues of the relevant blocks, which are Laplacians with periodic +1's on the diagonal; We show that the ground energy for blocks with incorrect *yes* guesses would then behave inversely with the square of the *period*'s length, and will thus be respectively large.

To penalize incorrect *no* guesses, we introduce two consecutive $+1/2$ penalty terms on the diagonal for *every* computation (even correct ones). We show that the ground energy of the resulting

block (where all *yes* guesses are correct) is exactly

$$1 - \cos(\pi/(L+1)), \tag{2}$$

which is a function of L. Our idea is that we can vary the *length of the computation* L to control the ground energy, in order to implement the energy penalty for the *no* guesses. This is achieved by defining the following function $T(x, y)$, for a given input x and oracle guess sequence y:

$$T(x, y) = f(x, y) + 2^m \cdot 4^{m+1} + 2^m \cdot \sum_{j=1}^{m} y_j \cdot 4^{m-j+1} \tag{3}$$

Note that a *yes* guess ($y_j = 1$) increases the value of T and in turn of L, and thus decreases the lowest eigenvalue. Therefore *no* guesses have an implicit cost. The function T has the required exponential structure so that the cost of a *no* guess decreases exponentially with each query. This fact, along with the fact that incorrect *yes* guesses incur a very high periodic cost, guarantees that the smallest eigenvalue will correspond to a block with $T(x, \tilde{y})$, where $\tilde{y}$ is the string of correct oracle responses.

The function $f(x, y)$ in the definition of $T(x, y)$ is the outcome of the computation on input x, when string y is used for the oracle responses made by the Turing Machine computing f. If $\tilde{y}$ is the set of correct oracle responses, then $f(x) = f(x, \tilde{y})$. We show in the proof of Theorem 2.17 (given in the full version [2]) that if the ground energy $(1 - \cos(\pi/(L+1)))$ can be computed to a 1/poly precision for a sufficiently high degree polynomial, then the value of $L = p(N) \cdot (2T(x, \tilde{y}) + 1)$ can be recovered, from which $T(x, \tilde{y})$ can be recovered. Here we are assuming that $f(x, y)$ is of length at most m; this can be guaranteed using a standard padding argument (see [2]). Thus, as can be seen from the expression for $T(x, y)$ given in Equation (3), $f(x, \tilde{y})$ is just the low order bits of $T(x, \tilde{y})$.

1.3 Proof Overview: The Infinite Case

We use the technique introduced by Cubitt, Perez-Garcia, and Wolf in [10] who incorporate an apreiodic tiling structure into the Hamiltonian term for the infinite plane. They use Robinson tiles [20] which are a finite set of tiling rules that when applied to the infinite plane, force an aperiodic structure with squares of exponentially increasing size. Each square has size 4^k, for positive integer k, and the density of squares of size 4^k in the limit of the infinite plane is $1/4^{2k+1}$. Tiling rules are essentially a classical version of local Hamiltonians, so the tiling rules can be encoded into a 2D layer of the Hilbert space on the plane. As was done in [10], we layer a TI 1D Hamiltonian on top of one of the sides of all the squares. The tiling pattern on the lower layer determines where the 1D term is applied. The effect of this structure is that the ground energy density α_0 for the infinite plane is the sum of the ground energies for an infinite series of finite 1D systems divided by the density of each square size:

$$\alpha_0 = \sum_{x=1}^{\infty} \frac{\lambda_0(N_x)}{4(N_x)^2}, \tag{4}$$

where $N_x = 4^{x^2}$ and $\lambda_0(N_x)$ is the ground energy of a 1D chain of length N_x with the Hamiltonian term from the 1D finite construction applied to each pair in the chain. Note that it is essential here that the TI construction for finite 1D chains have a fixed Hamiltonian term that does not depend on the chain size since in the infinite construction the same Hamiltonian term is applied to every pair of neighboring particles along each dimension.

As noted earlier, our reduction needs to encode an entire language in one, infinite-precision number α_0. In our construction, different portions of the binary representation of α_0 encode the value of a function f on different inputs. In particular, bits $4x^2$ through $4(x+1)^2$ encode the value of $f(x)$. The top segment of each square of size 4^{x^2} in the Robinson tiling of the infinite plane is layered with a TI 1D Hamiltonian whose ground energy encodes the value $f(x)$, where the function problem f is from the oracle complexity class $\mathrm{FEXP}^{\mathrm{NEXP}}$. This 1D Hamiltonian is exactly the construction used in the finite case, except that the computation is in FEXP instead of FP as in the finite case. The reason for the exponential increase in complexity is that we are using a square of size 4^{x^2} to encode the computation on input x.

Each $\lambda_0(N_x)$ in Equation (4) is an irrational number, so bits $4x^2$ through $4(x+1)^2$ which encode the value of $f(x)$ will also include the "overflow" from the energy contributions of $f(x')$ for every $x' < x$. The values of f on inputs $x' < x$, then need to be calculated to the required precision in the classical calculation of the reduction and subtracted off from α_0 in order to recover the bits required to reconstruct the value of $f(x)$. Using the analysis of our construction for the finite 1D case, we know that $\lambda_0(N_x) = (1 - \cos(\pi/(L_x+1))$, where L_x is the integer equal to $p(N_x) \cdot (2T(x, \tilde{y}) + 1)$. Recall that $T(x, \tilde{y})$ encodes the desired value for $f(x)$.

We now sketch how the first $4(x+1)^2 + 2$ bits of α_0 are sufficient to recover $\lambda_0(N_x)$ by inductively subtracting off $\lambda_0(N_{x'})/4(N_{x'})^2$ for every $x' < x$. Note that dividing $\lambda_0(N_{x'})$ by $4(N_{x'})^2$ effectively shifts the binary representation of $\lambda_0(N_{x'})$ to the right by $\log_2[4(N_{x'})^2] = 4(x')^2 + 2$ bits. Let $\overline{\alpha}_0$ be the current value of the sum. Initially, $\overline{\alpha}_0$ is equal to the first $4(x+1)^2 + 2$ bits of α_0. In an inductive step, one has subtracted off $\lambda_0(N_{x'})/4(N_{x'})^2$ for every $x' < z$ for some $z \leq x$. As shown in Figure, 1, the first $4z^2 + 2$ bits of $\overline{\alpha}_0$ are 0 and the next $4(z+1)^2 - 4z^2$ bits are determined solely by $\lambda_0(N_z)$. We show in the proof of Theorem 2.17 that this is enough information to recover the integer L_z where $\lambda_0(N_z) = (1 - \cos(\pi/(L_z+1))$. Once L_z is recovered, $\lambda_0(N_z)$ can be computed to any desired accuracy. In particular, it can be calculated up to $4(x+1)^2+2$ bits of precision and subtracted off from $\overline{\alpha}_0$. The result is a new $\overline{\alpha}_0$ which is equal to the sum $\sum_{x'=z+1}^{x} \lambda_0(N_{x'})/4(N_{x'})^2$ up to $4x^2 + 2$ bits of precision. This illustrates why it is necessary to calculate the ground energy for each finite chain to an arbitrary level of precision: for every $z < x$, it is necessary to calculate $\lambda_0(N_z)$ to a precision of $4x^2+2$ bits in order to derive $f(x)$ from a $4x^2+2$-bit approximation of α_0.

1.4 Summary of Main New Technical Contributions in the Proof

While the results in the paper make use of many previously known techniques from Hamiltonian complexity, there are several novel technical contributions introduced here which we hope will be useful elsewhere.

As mentioned above, to the best of our knowledge this is the first time that the length of the computation (which we control using

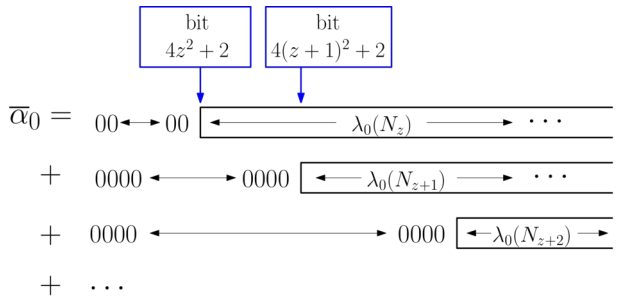


Figure 1: The partial sum $\overline{\alpha}_0$. The first $4z^2+2$ bits of $\overline{\alpha}_0$ are 0 and the next $4(z+1)^2-4z^2$ bits are determined solely by $\lambda_0(N_z)$. This is enough information to recover the integer L_z where $\lambda_0(N_z) = (1-\cos(\pi/(L_z+1))$.

the number of repetitions $T(x,y)$) is used to control the ground energy and make sure that the ground value indeed corresponds to the correct initial parameters.

We believe this is also the first time in Hamiltonian complexity that such precise control over the eigenvalues was required. We need the high level of precision in several places in the proof. First, in the infinite case, the precise analysis of the lowest eigenvalue is essential in simultaneously encoding an infinite sequence of numbers (values $f(x)$ for an infinite sequence of x's) into a single, infinite-precision number. The high level of precision is needed so that given a precise enough estimate for α_0 we can inductively subtract off the energy contributions of $f(x')$ for each $x' < x$, in the classical computation of the reduction; this then leaves enough information to determine $f(x)$.

Secondly, even in our finite case construction, the fact that we are considering a function problem implies that we must distinguish between many different values for the cost function as opposed to extracting only a single bit as required for a decision problem. This requires a more precise handle on the lowest eigenvalue of the resulting matrix.

Last but not least, in the finite case, as explained in subsection 1.3, we use highly precise lower bounds to show that the energy penalty for incorrect *yes* oracle guesses, which we achieve by periodic penalties of period $p(N)$, is indeed larger than the ground energy in the case that all *yes* guesses are correct. The usual lower bound of $\Omega(1/p(N)^3)$ used in previous circuit-to-Hamiltonian constructions does not suffice, as it needs to be larger than our upper bound on the ground value, which is $O(1/T^2P(N)^2)$ by Equation (2). We cannot take T to be arbitrarily large, since T has to be computed in unary in less than N steps and so it is bounded by roughly $\sqrt{N}$.

The use of the circular clock also requires some additional care, as it is important that the computation part of the state also returns to its initial state at the end of the cycle. By contrast, if the computation is a path, the ending state of the computation can be arbitrary. We handle this by embedding a computation which executes a process in the forward direction for a certain number of steps. Since the action on the computation tracks is reversible, the computation can be undone by executing the same number of reverse steps.

1.5 Concluding Remarks, Related Work and Open Questions

To our knowledge, this is the first time the complexity of calculating physical quantities in the TL for a fixed Hamiltonian is characterized from a computational perspective. To this end we define a function problem which captures roughly the complexity of the task that a physicist encounters when attempting to calculate a physical quantity with one more bit of precision; we show, roughly, that it is doubly exponential. To our knowledge this is also the first use of a circuit-to-Hamiltonian construction to show hardness for a *function* problem.

After completing this work, we learned that Cubitt and Watson independently considered the problem of computing the ground energy density in the TL with a fixed Hamiltonian term [22]. They study a classical two-threshold version of our GED problem (which they call GSED), and prove that the class of languages computable by an exponential time Turing Machine with access to GSED is contained in $\mathrm{EXP}^{\mathrm{NEXP}}$ and contains P^{NEEXP}. Their hardness result thus holds for Turing reductions but not for Karp reductions, as in our reduction; using Turing reductions frees them from the need to employ Krentel's accounting scheme to embed the entire interaction with an oracle in a single instance, which is the primary technical contribution in our work. In their proof, the finite construction embedded in each square of the Robinson tiling is a NEEXP computation, as opposed to the $\mathrm{FEXP}^{\mathrm{NEXP}}$ computation implemented in our finite construction. We note that we do not know how to prove Karp reduction results for the two-threshold problem variant. The problem is that in a Karp reduction, a polynomial time reduction must take an instance x of a language L and create an instance of GSED that encodes whether $x \in L$. The instance of GSED must, by definition, include thresholds and computing the correct threshold for the n^{th} bit of α_0 essentially requires computing the first $n-1$ bits of α_0.

Importantly, our results are not tight. We note that in our hardness results, since we are reducing from functions $f \in \mathrm{FP}^{\mathrm{NEXP}}$, and $f \in \mathrm{FEXP}^{\mathrm{QMA\text{-}EXP}}$, the verifier V and hence all of the computations we are simulating are classical. We still need a quantum construction to execute the clock and create a ground state that is a superposition of the states of the computation at each clock step. We conjecture that the finite problem is in fact hard for $\mathrm{FP}^{\mathrm{QMA\text{-}EXP}}$, and the infinite one is hard for $\mathrm{FEXP}^{\mathrm{QMA\text{-}EXP}}$. We believe that our constructions can be generalized in a straight-forward way to encode the computation of quantum verifiers and quantum witnesses required for a QMA-EXP oracle instead of the classical ones required for NEXP. The main problem in completing the quantum hardness result is that QMA and QMA-EXP are classes of promise problems which means that the Turing Machine can make invalid oracle queries, and there is no guarantee on the responses for invalid queries. Recently, Gharibian and Yirka [14] and later Watson, Bausch, and Gharibian [21] managed to get around the invalid queries issue in the finite APX-SIM problem, for quantities other than the ground-energy; however, in the context of measuring the ground energy itself, it is unknown how to do this since invalid queries have an uncontrolled effect on the ground energy. As written, the constructions in [21], while TI, also have input-dependent

Hamiltonian parameters, but this could probably be fixed. The issue of invalid queries, on the other hand, appears to be an obstacle for extending these results to the ground energy, even when the Hamiltonian terms are position-dependent as in [13, 14], as well as in the finite TI constructions given here and in [21]. It remains open to provide tight complexity bounds for computing the ground energy to within a given precision, even in the finite non-TI case. Another interesting open problem is to extend the works of [21] to the infinite case, and arrive at *tight* complexity results in the TL fixed Hamiltonian case, for measuring quantities other than energy.

We note that several recent works [5, 6, 9, 10] study a relaxed notion of TI (called *semi*-TI in [3]), in which the Hamiltonian is a weighted sum of fixed terms; the weights are given as part of the input. This notion seems less natural for the TL setting where infinitely many weights need to be given in order to specify the problem.

2 COMPUTING GROUND ENERGIES FOR FINITE TI HAMILTONIANS

In this section we provide more details for the main technical proof in the paper which is that computing the ground energies of finite tranlsational-invariant Hamiltonians is hard for $\mathrm{FP}^{\mathrm{NEXP}}$. We refer the reader to the full version [2] for a complete description of the construction and proofs.

2.1 Eigenvalue Bounds

The analysis of the construction requires new bounds on the smallest eigenvalues of certain matrices. The proofs of the second and third bounds are given in the full version [2].

Let C_L be the propagation matrix for a matrix for a cycle of length L. Note that C_L is $1/2$ times the Laplacian matrix for a cycle of length L. We will use P_L to denote the $L \times L$ matrix which is $1/2$ times the Laplacian matrix for a path of length L.

Lemma 2.1. **[Smallest Eigenvalue for the Path Graph Plus a Penalty]** *Let D_1 be the matrix that is all 0's, except for a 1 in the upper left corner. The smallest eigenvalue for $P_L + D_1$ is at least*

$$\left(1 - \cos\left(\frac{\pi}{2L+1}\right)\right)$$

Proof. [11] show that the smallest eigenvalue of $P_L + \frac{1}{2}D_1$ is exactly

$$\left(1 - \cos\left(\frac{\pi}{2L+1}\right)\right).$$

Since the D_1 is positive semi-definite, the smallest eigenvalue of $P_L + \frac{1}{2}D_1$ is a lower bound for the smallest eigenvalue of $P_L + D_1$. The lemma follows. □

Lemma 2.2. **[Smallest Eigenvalue for the Cycle Graph Plus Two $1/2$ Penalties]** *Let $D_{2\times\frac{1}{2}}$ be an $L\times L$ matrix that is 0 everywhere except for two consecutive diagonal entries which are $1/2$. The smallest eigenvalue of $C_L + D_{2\times\frac{1}{2}}$ is exactly*

$$\left(1 - \cos\left(\frac{\pi}{L+1}\right)\right).$$

Lemma 2.3. **[Smallest Eigenvalue Lower Bound for the Cycle Graph Plus a Periodic Penalty]** *If $L = rs$ and let $D_{periodic}$ be a diagonal matrix which is has all zero entries except for $D_{periodic}[i,i] = 1$ if i is an integer multiple of s. The smallest eigenvalue of $C_L + D_{periodic}$ is at least*

$$\frac{1}{8}\left(1 - \cos\left(\frac{\pi}{2s+1}\right)\right)$$

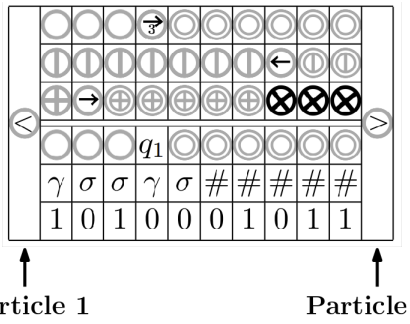


Figure 2: Sample configurations for the clock and computation tracks. The clock tracks are the top three tracks (Tracks 1, 2, and 3) and the computation tracks are the bottom three tracks (Tracks 4, 5, and 6). The computation tracks represent the configuration of a Turing Machine with Track 4 representing the state and location of the head, Track 5 representing the contents of the work tape, and Track 6 representing a read-only witness string.

2.2 The Track Structure of the Hilbert Space

The Hilbert space of each particle is a tensor product of 6 different spaces which will be used to form 6 tracks. There are also two additional states ⊴ and ⊵. So the Hilbert space for particle i is

$$\mathcal{H}_i = \{\trianglelefteq, \trianglerighteq\} \oplus \left(\mathcal{H}_{i,1} \otimes \mathcal{H}_{i,2} \otimes \mathcal{H}_{i,3} \otimes \mathcal{H}_{i,4} \otimes \mathcal{H}_{i,5} \otimes \mathcal{H}_{i,6}\right).$$

Later, we will introduce constraints that enforce the condition that for a low energy state, the leftmost particle must be in state ⊴, the right particle must be in state ⊵ and the particles in between are not in states ⊴ or ⊵. We will call all such standard bases states *bracketed*. The space spanned by all bracketed states is $\mathcal{H}_{br}$ and the final Hamiltonian is invariant on $\mathcal{H}_{br}$. For any bracketed state, Track k of the system of N particles consists of the tensor of $\mathcal{H}_{i,k}$ as i runs from 2 to $N-1$.

Three of the tracks are *clock* tracks which contain the state of the clock. The other tracks are *computation* tracks that encode the configuration of a Turing Machine computation. An example is given in Figure 2. The symbols in the figure above represent standard basis states for each portion of the Hilbert space.

2.3 Circuit-to-Hamiltonian Preliminaries

The Hamiltonian term h in the construction is a sum of two body terms, and there are two types of terms. Type I terms will have the form $|ab\rangle\langle ab|$ where a and b are possible states. This has the effect of adding an energy penalty to any state which has a particle in state a to the immediate left of a particle in state b. In this case, we will refer to ab as an *illegal pair*. We will sometimes consider the restriction of an illegal pair to a set of tracks. In this case we implicitly mean that the Hamiltonian term acts as the identity on the remaining (unspecified tracks). The restriction of an illegal pair

to a set of tracks will sometimes be referred to as an *illegal pattern*. We will refer to standard basis states of the whole chain or a subset of the tracks as *configurations*. Any configuration which has an illegal pattern or illegal pair is said to be illegal.

Type II terms will have the form: $\frac{1}{2}(|ab\rangle\langle ab|+|cd\rangle\langle cd|-|ab\rangle\langle cd|-|cd\rangle\langle ab|)$. These terms enforce that for any eigenstate with zero energy, if there is a configuration A with two neighboring particles in states a and b, there must be a configuration B with equal amplitude that is the same as A except that a and b are replaced by c and d. Even though a Type II term is symmetric, we break this symmetry and associate with each such term a direction by choosing an order on the pair of pairs. If we decide that ab precedes cd, then we denote the term as: $ab \rightarrow cd$. Type II terms are also referred to as *transition rules*. We will say that configuration A transitions into configuration B by rule $ab \rightarrow cd$ if B can be obtained from A by replacing an occurrence of ab with an occurrence of cd. We say that the transition rule applies to A in the forward direction and applies to B in the backwards direction. The sum of all the Type II terms is called h_{prop} or the *propagation Hamiltonian*. Again, we may specify that a transition rule acts on a particular subset of the tracks: $ab_T \rightarrow cd_T$, where T is a subset of the tracks. This implicitly means that the Hamiltonian term acts as the identity on the remaining tracks. It will also sometimes be convenient to say that a transition rule applies to every state except a particular standard particle basis state. In this case, we will denote the rule by: $a(\neg b) \rightarrow cd$ to indicate that the rule applies to every ax pair, where $x \neq b$.

2.4 The Clock

In this subsection we focus on the Hamiltonian restricted to the subspace of the clock. The clock used in this construction is a more elaborate version of the clock used in [15]. Any standard basis state of Tracks 1 through 3 is called a *clock configuration*. We define a set of *well formed* clock configurations such that the condition of being well formed can be enforced by Type I terms. Thus, all clock configurations which are not well formed will have an energy penalty and do not appear in the support of the ground state. Let h_{wf-cl} denote the Hamiltonian terms with the constraints that give an energy penalty for any clock state that is bracketed but not well-formed.

The well-formed clock configurations have a special structure defined by a configuration graph:

Definition 2.4. **[Configuration Graph]** *The vertices of the* configuration graph *correspond to the well-formed clock configuration for Tracks* 1, 2, *and* 3. *There is a directed edge from configuration c_1 to configuration c_2, if c_2 can be reached from c_1 by the application of one transition rule in the forward direction.*

We prove the following lemma which establishes that the configuration graph is a set of disjoint paths and cycles.

Lemma 2.5. **[Degree Bound for the Configuration Graph]** *The transition rules are closed on the set of well-formed clock configurations. Furthermore, for every well-formed clock configuration, there is at most one transition rule that applies in the forward direction and at most one transition rule that applies in the reverse direction.*

We first describe the structure of the configuration considering just the configurations for Tracks 1 and 2 and then further refine the analysis with Track 3 added in.

Track 1 has exactly one pointer that shuttles back and forth between the left and right ends of the chain according to the transition rules. The ⊖ pointer moves to the right and the ⊖ pointer moves to the left. The Track 1 pointers come in 8 different varieties and are labeled with tags in the range from 1 through 8. There are two pointers with the label i: ⊖ and ⊖. The different types of Track 1 pointers act as a means of program control as they trigger different operations on the three computation tracks (Tracks 4, 5, and 6).

In each iteration for Tracks 1 and 2, two different Turing Machines (described in the next subsection) are run sequentially, each for $N-2$ steps. The Track 1 pointer acts as a second hand for the clock. In a round trip of a Track 1 pointer, a single step of the Turing Machine is executed on the computation tracks. The number of Turing Machine steps in the iteration is regulated by the pointer in Track 2 which is advanced by one location as the Track 1 pointer sweeps by. Thus, the Track 2 pointer acts as a minute hand for the clock. Some of the Track 1 pointers act as the identity on the computation tracks and are used to check certain local conditions and impose an energy penalty if those conditions are not met.

We further classify well-formed configurations for Tracks 1 and 2 as either *correct* or *incorrect*. Type I constraints are added which will ensure that incorrect clock configurations will never be in the support of the ground state. The complete specification of these Type I terms is given in the full version and the sum of these terms is called h_{cl}.

Lemma 2.6. **[Incorrect Configurations form Short Paths]** *Any incorrect clock configuration for Tracks* 1 *and* 2 *will reach an illegal configuration (from h_{cl}) in at most* $2(N-2)$ *clock steps from which there is no out-going transition.*

There are exactly $p(N) = 4(N-2)(2N-3)$ correct clock configurations for Tracks 1 and 2. The following lemma establishes that the correct clock configurations for Tracks 1 and 2 form a single cycle:

Lemma 2.7. **[Correct Configurations Form Cycles]** *Starting from any correct configuration c for Tracks* 1 *and* 2, *the transition rules will reach every correct configuration for Tracks* 1 *and* 2 *exactly once before returning to configuration c after* $p(N) = 4(N-2)(2N-3)$ *clock steps.*

We will refer to a sequence of $p(N)$ clock steps as a *iteration*.

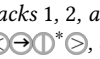
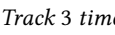
The third clock track acts like an hour hand in that the pointer on Track 3 moves by one location each time an iteration is completed. Track 3 is used to hold a *timer*, that will count the number of iterations. The number of ⊕ or ⊕ particles, denoted by T, is called the *timer length* for Track 3, and the transition rules never alter the length of the timer. If the timer length is T, the iteration will be repeated $2T+1$ times, to form a *clock cycle*. This is done as follows.

The Track 3 pointer shuttles back and forth, as do the pointers on Tracks 1 and 2. However, the right-moving Track 3 pointer will turn around when it reaches the the left-most ⊗ particle. This provides a way to control the number of iterations of Tracks 1 and 2 in each clock cycle. In the sample configuration shown in Figure 2, the Track 3 timer has length 6. If the length of the timer is T, then

Track 3 transitions through $2T+1$ configurations before repeating. Referring again to the example in Figure 2, since the timer has length 6, there will be 13 iterations of Tracks 1 and 2 for each clock cycle. Starting in a correct clock configuration in which Track 3 has a timer length T, after $(2T+1)\cdot p(N)$ time steps, the clock configuration will return to its original state completing a clock cycle. We note that in referring to a *clock configuration* without specifically restricting to a subset of the tracks, we are referring to a configuration for all three clock tracks.

The notion of correct and incorrect clock configurations is generalized to Tracks 1, 2, and 3 by adding additional constraints on Track 3. These additional constraints are also added to the h_{cl} term. The structure of the configuration graph then is summarized in the following lemma:

Lemma 2.8. **[Structure of the Configuration Graph]** *In the configuration graph, the correct clock configurations with timer length T form a cycle of length $(2T+1)p(N)$. Every correct clock configuration for Tracks 1 and 2 occurs exactly $2T+1$ times in the cycle at intervals of exactly $p(N)$. All other well-formed clock configurations are in paths of length at most $p(N)$ that end in an illegal configuration (from h_{cl}).*

Definition 2.9. **[Enumeration of Correct Clock Configurations]** *A clock configuration is at Time 0 for Tracks 1, 2, and 3 if it's Track 1 state is*  , *it's Track 2 state is* , *and its Track 3 state is*  .

$|c_{T,0,0}\rangle$ is the clock configuration in which the Track 3 timer has length T and Tracks 1, 2, and 3 are all at Time 0. Note that the timer length is well-defined for $T \in \{0,\ldots,N-3\}$. For $s \in \{0,\ldots,2T\}$ and $t \in \{0,\ldots,p(N)-1\}$, $|c_{T,s,t}\rangle$ is the clock configuration that is reached after $s\cdot p(N)+t$ clock steps starting in $|c_{T,0,0}\rangle$. s uniquely determines the configuration of Track 3 and t uniquely determines the configuration of Tracks 1 and 2.

Lemma 2.12 (proven in Section 2.9.1) uses this structure to show that the space spanned by incorrect clock configurations have high energy. The Hamiltonian restricted to correct clock configurations is block diagonal, with one block corresponding to each possible timer length T. The block corresponding to timer length T is $1/2$ times the Laplacian of a cycle graph with $(2T+1)\cdot p(N)$ vertices. When the Hilbert space for the computation tracks is tensored with the Hilbert space for the clock, the state space expands but the Hamiltonian retains this block diagonal structure. In the expanded space that includes both the computation and the clock space, every block is parametrized by its timer length T in addition to the initial configuration of the Turing Machine, denoted by v-init (describing the state of Tracks 4 and 5) as well as the read-only witness string on Track 6. It will be convenient to divide the witness string into a string y used as the guesses for the responses to the oracle queries and w for the witness string used in simulating the verifier for the calls made to the oracle in which the guess is *yes* (for each such *yes* guess there will be a different witness string w_i). Thus, every block is parameterized by the 4-tuple $(T, v\text{-init}, y, w)$. Penalty terms described in Subsection 2.8 will be added in to ensure that blocks corresponding to incorrect parameters have a high ground energy. First we describe in Subsection 2.5 the computation performed in each iteration.

2.5 Overview of the Computation Embedded in the Hamiltonian

Here we give an overview of the computation used in the construction. This computation is repeated $2T+1$ times.

Stage 1 - Binary Counter Turing Machine: A Turing Machine that increments a binary counter called M_{BC} is run for $(N-2)$ TM steps, where N is the length of the chain. If the starting configuration has the string "1" on the work tape, then after $(N-2)$ TM steps, there will be some string x on the work tape representing the number of increment operations performed. x is then used as the input to the next stage of the computation.

The idea of using a binary counter TM to translate the length of the 1D chain into an input string x was used in [15] as well. Define $N(x)$ to be the number such that after $(N(x)-2)$ TM steps, the binary counter TM completes an increment operation with x on the work tape. The function mapping x to $N(x)$ is the reduction. The complete specification of the binary counter Turing Machine as well as an explicit formula for $N(x)$ is given in the full version [2].

Stage 2 - Timer and Verification Turing Machine: In the second stage of the computation, a Turing Machine called M_{TV} simulates the verifier for oracle responses for which the guess response is *yes*, and then computes the function $T(x,y)$ shown in Equation (3). The first m bits of the witness track are used as the guess y for the oracle responses and the remaining bits w are used as witnesses for the verifier as needed. If $f \in \mathrm{FP}^{\mathrm{NEXP}}$, then f is computed by a polynomial time Turing Machine M with access to an oracle for language $L \in \mathrm{NEXP}$. The exponential time verifier for L is called V. First M is simulated on input x using the oracle responses given by y. This completely determines the oracle inputs $x_1,\ldots,x_m$. If y guesses that an oracle response is *yes* ($y_i=1$) then V is simulated on input x_i using bits from the witness w from the witness track. If any of the oracle computations that correspond to *yes* guesses are rejecting, a character is placed on the work tape that will trigger a penalty term at the second checking phase, which occurs at the end of the computation (as explained in Subsection 2.8). Next the function $T(x,y)$ is computed and written in unary on Track 3 so that it can be checked (again, in the second checking phase) against the timer used to regulate the number of iterations of the clock. The pseudo-code for the computational process executed in the second $N-2$ TM steps is given in Figure 3.

The following definition allows us to refer to the correct output of M_{TV} whose behavior depends on the input pair (x,ω) as well as Turing Machines M and V. ω is the binary witness string ω of length $N-2$ written on the witness track and contains the string y used for oracle responses as well as the witnesses w used in verification.

Definition 2.10. *Define* $\mathrm{Out}(x,\omega,M,V)$ *as the correct output of M_{TV} on input (x,ω) using Turing Machines M and V, as described in Figure 3.*

INPUT: (x, ω)
(1) Compute $N = N(x)$
(2) m is the number of oracle queries made by M on input x
(3) r is the size of the witness used by V on any oracle query generated by M on input x
(4) m and r are hard-coded functions of $|x|$ determined by M and V.
(5) Simulate Turing Machine M on input x
(6) Use y for the responses to the oracle queries, where y denotes the first m bits of ω
(7) Simulation generates $x_1, \ldots, x_m$, inputs to oracle queries
(8) REJECT = FALSE
(9) **for** $i = 1 \ldots m$
(10) **if** $y_i = 1$
(11) Simulate V on input (x_i, w_i)
(12) w_i is the string formed by bits $m + (i-1)r + 1$ through $m + ri$ of ω
(13) If V rejects on input (x_i, w_i)
(14) REJECT = TRUE
(15) **if** (REJECT)
(16) Write σ_R in the left-most position of the work tape
(17) **else**
(18) Write σ_A in the left-most position of the work tape
(19) Compute $T(x, y)$
(20) Write the value of $T(x, y)$ in unary with σ_X symbols starting in the second position of the work tape
OUTPUT: $(\{\sigma_A, \sigma_R\}(\sigma_X)^{T(x,y)}; x, \omega)$

Figure 3: Pseudo-code for the Turing Machine M_{TV}.

2.6 Implementing TM Steps in Propagation Terms

We will add Type I terms that will give an energy penalty to any standard basis state for the computation tracks that do not correspond to a well-defined Turing Machine configuration. These are the well-formed computation configurations. Let h_{wf-co} denote the Hamiltonian terms with the constraints that give an energy penalty for any state of the computation tracks that is not well-formed. C_N is defined to be the set of standard basis states for the well-formed computation configurations.

A single sweep of a pointer on Track 1 ⟵ from right to left causes the TM configuration on Tracks 4 through 6 to advance by one TM step. The details of this part of the construction follow [15] closely and are given in the full version [2]. Note that since we are reducing from FP^{NEXP} instead of $\text{FP}^{\text{QMA-EXP}}$, all of the Turing Machines in our construction, including the verifier V, are classical reversible Turing Machines instead of Quantum Turing Machines.

Define S to be the set of standard basis states associated with the computation tracks of two neighboring particles. The propagation terms in h_{prop} for a forward step of a TM all have the form:

$$\frac{1}{2}[I_S \otimes |\leftarrow\bigcirc\rangle\langle\leftarrow\bigcirc| + I_S \otimes |\bigcirc\leftarrow\rangle\langle\bigcirc\leftarrow|$$
$$+ \quad P \otimes |\leftarrow\bigcirc\rangle\langle\bigcirc\leftarrow| + P^\dagger \otimes |\bigcirc\leftarrow\rangle\langle\leftarrow\bigcirc|]$$

P is a unitary operation that applies to the space spanned by S. In fact since the computations are classical the function P is a permutation on S. Each of the two Turing Machines M_{BC} and M_{TV} give rise to two different permutations P_{BC} and P_{TV}.

As the Track 1 pointer sweeps from right to left, P is applies to each pair of particles from right to left. The particles will be numbered from left to right 0 through $N-1$, so the non-bracketed particles are 1 through $N-2$. Applying P to particles i and $i+1$ will be denoted $P^{(i,i+1)}$. The following sequence of operations accomplishes one forward step of the Turing Machine associated with P:

$$P^{(1,2)} P^{(2,3)} \ldots P^{(N-3,N-2)}$$

The reverse step a Turing Machine is then accomplished with a left to right sweep of a Track 1 ⟶. These propagation terms have the form:

$$\frac{1}{2}[I_S \otimes |\rightarrow\bigcirc\rangle\langle\rightarrow\bigcirc| + I_S \otimes |\bigcirc\rightarrow\rangle\langle\bigcirc\rightarrow|$$
$$+ \quad P \otimes |\rightarrow\bigcirc\rangle\langle\bigcirc\rightarrow| + P^\dagger \otimes |\bigcirc\rightarrow\rangle\langle\rightarrow\bigcirc|]$$

As the Track 1 pointer sweeps from left to right, $P^\dagger$ to each pair. A single sweep across the entire chain applies the operations:

$$(P^{(N-3,N-2)})^\dagger \ldots (P^{(2,3)})^\dagger (P^{(1,2)})^\dagger$$

2.7 Overview of an Iteration

We will use the different i-pointers on Track 1 to trigger different operations, depending on whether the operation is a forward or reverse step and whether the Turing Machine step is dictated by M_{BC} or M_{TV}. The list below shows the segments for an entire iteration for Tracks 1 and 2 and how many steps of each Turing Machine are triggered on the computation tracks during the segment. Segments 1 through 3 trigger a step of one of the two TMs as the 1-pointer moves from right to left. A sweep of the ⟵ triggers the application of the corresponding P operation to each pair of particles from right to left. Segments 5 through 7 trigger the inverse operation of the TMs. A sweep of the ⟶ triggers the application of the corresponding $P^\dagger$ operation to each pair of particles from left to right. This will cause the state of the computation tracks to return to their original state after each iteration for Tracks 1 and 2. Segments 4 and 8 are used for checking purposes only. The 4 and 8-pointers act as the identity on the computation tracks. We will add Type I terms that cause a penalty in the presence of a 4 or 8-pointer to penalize certain conditions on the computation tracks described in the next subsection.

1) N-3 forward steps of M_{BC}
2) One forward step of M_{BC}
3) N-2 forward steps of M_{TV}
4) Identity
5) N-2 reverse steps of M_{TV}
6) One reverse step of M_{BC}
7) N-3 reverse steps of M_{BC}
8) Identity

Figure 4: The Turing Machine steps executed in each i-segment.

When the forward steps of M_{BC} and M_{TV} are executed, P_{BC} is applied to each pair of neighboring particles, from left to right $N-2$ times, followed by P_{TV} applied to each pair of neighboring particles, from left to right $N-2$ times. When the reverse steps of M_{BC}

and M_{TV} are executed, $P_{TV}^{\dagger}$ is applied to each pair of neighboring particles, from right to left $N-2$ times, followed by $P_{BC}^{\dagger}$ applied to each pair of neighboring particles, from right to left $N-2$ times. The net effect is that the the computation tracks return to the same state at the end of each iteration for Tracks 1 and 2 as formerly stated in the lemma below. The notation $|c_{T,s,t}\rangle$ refers to the enumeration of correct clock configurations given in Definition 2.9.

Lemma 2.11. **[The Computation is Cyclic]** *Consider a sequence of clock steps from $|c_{T,s,p(N)-1}\rangle$ to $|c_{T,s+1,p(N)-1}\rangle$. If the computation tracks start out in a state $|\phi\rangle$, then the state of the computation tracks is unchanged at the end of the sequence of clock steps. That is, $p(N)$ clock steps applied to $|c_{T,s,p(N)-1}\rangle|\phi\rangle$ will result in the state $|c_{T,s+1,p(N)-1}\rangle|\phi\rangle$*

2.8 Penalty Terms and Lowest Eigenvalues

We now explain the checking phases, the conditions they impose, and how those conditions determine the lowest energy eigenvalue. If any of the conditions enforced in the checking phases are violated, that will trigger a penalty (via Type I terms) that occurs every $p(N)$ clock steps for a total of $2T+1$ times during one full iteration of the cycle of length $(2T+1)\cdot p(N)$. This will result in a block matrix corresponding to a cycle of length $L=(2T+1)\cdot p(N)$ with an additional +1 every $p(N)$ locations along the cycle. We use techniques from spectral graph theory to exploit the exact periodic structure of the penalty terms to show in Lemma 2.3 that the lowest energy eigenvalue of this matrix is at least

$$\frac{1}{8}\left(1-\cos\left(\frac{\pi}{2p(N)+1}\right)\right)=\Theta\left(\frac{1}{(p(N))^2}\right). \tag{5}$$

This lower bound is large enough to eliminate any blocks with these periodic costs. We now explain how we use this mechanism to create the large costs that penalize an incorrect *yes* guess for an oracle response as well as to check that the initial configuration and the timer lengths are correct.

The first checking phase takes place before the forward execution of the Turing Machines and verifies that the Turing Machine is starting in the correct initial configuration for Tracks 4 and 5 which is called v-init and is depicted below:

The Type I terms associated with this penalty are called h_{init}. If the computation begins in this special initial configuration then at the end of $N-2$ steps of the Turing Machine M_{BC}, the correct input x for chain length N will be written on the work tape.

The second checking phase occurs after the forward computation of the two Turing Machine and checks that the timer on Track 3 matches the calculation of $T(x,y)$ performed by M_{TV}. Thus, there will be a periodic cost unless the timer length T is equal to $T(x,y)$ for the correct input x and the y written on the witness track. The Type I terms associated with this penalty are called h_{length}. In addition, the second checking phase is used to check that all the simulations of the verifier performed by M_{TV} are accepting. In other words, a penalty is applied if any of the verifier computations were rejecting. The Type I terms associated with this penalty are called h_V.

Thus, if a block parameterized by $(T, v\text{-init}, y, w)$ does not have a periodic cost, then the computation generates the correct input x, the timer T is equal to $T(x,y)$, and all of the oracle guess where $y_i=1$ (*yes* guesses) are correct and use a correct witness from w in the verifying computation.

Finally, there is a penalty even for correct computations and clock configurations for every cycle for Tracks 1, 2, and 3. This term is called h_{final} and adds a penalty of $1/2$ to two consecutive clock states in every block.

The final 2-particle Hamiltonain term is:

$$h=h_{prop}+h_{wf-cl}+h_{wf-co}+h_{cl}+h_{init}+h_{length}+h_V+h_{final}.$$

For each of these terms h_*, we will use $H_{N,*}$ to refer to the Hamiltonian on a chain of length N obtained by applying the 2-particle term h_* to each pair of neighboring particles in the chain.

Now we discuss the mechanism used to apply the more modest penalty for *no* guesses for oracle responses. In our construction, every computation, regardless of the outcome of the computation, will incur two consecutive penalties of $+1/2$ at a particular point in the cycle. The resulting matrix is the propagation matrix for a cycle of length $L=(2T+1)\cdot p(N)$ with an additional $+1/2$ at two consecutive locations on the diagonal. We show in Lemma 2.2 that the lowest eigenvalue for this matrix is exactly $1-\cos(\pi/(L+1))$. Thus any block which does not have a periodic cost will have a smallest eigenvalue equal to $1-\cos(\pi/(L+1))$, with $L=(2T(x,y)+1)\cdot p(N)$. The block with the smallest eigenvalue will correspond to the y that has the largest value of $T(x,y)$ subject to the condition that y does not include any incorrect *yes* guesses. The exponential structure of the function $T(x,y)$ shown in Equation (3) guarantees that the block with the smallest eigenvalue uses the correct oracle guesses $\tilde{y}$, as shown in Lemma 2.15.

Recovering the value of $f(x)$ from an approximation of the ground energy requires a much sharper analysis of the lowest eigenvalue than has been required in previous constructions.

2.9 Analysis of the Ground Energy of the Hamiltonian

This section contains the analysis of the ground energy of the Hamiltonian and a proof that if the chain length is $N(x)$, then the value $f(x)$ can be determined by a sufficiently accurate $(1/poly(N))$ estimate of the ground energy.

Section 2.9.1 contains the main analysis of the ground energy of the Hamiltonian. The Hamiltonian has a block-diagonal structure which allows us to analyze the energy of each block independently. We first eliminate blocks that correspond to incorrect clock states. Then we eliminate blocks in which the initial configuration of the computation is incorrect. Thus, we can assume that we have a correct computation which uses the correct string x as the input. A block is also parameterized by y the first m bits of the witness tape which are used as guesses for the responses of the oracle. We eliminate blocks in which the timer length on Track 3 does not correspond to the correct $T(x,y)$ and blocks in which y contains an incorrect *yes* guess. These all have periodic costs that occur at least once every $p(N)$ clock steps in an iteration. Finally, we give an exact expression for the lowest eigenvalue within blocks which do not have a periodic cost, and show that blocks that correspond

to the correct y will have the smallest eigenvalue. Then we show how the value of $T(x,y)$ and $f(x)$ can be extracted from a good approximation of this smallest eigenvalue.

Section 2.9.2 finally addresses the assumption we have been making all along that the state is in the span of all bracketed states. Additional terms are added to the Hamiltonian which ensure that the ground state of the Hamiltonian is in fact in the span of all bracketed states. Theorem 2.17 ties all the various pieces of the finite construction together and gives the final result on the hardness of Function-TIH.

2.9.1 Analysis of the Ground Energy. Let $\mathcal{H}_{wf}$ denote the span of the bracketed well-formed clock configurations tensored with the span of C_N, the set of well-formed computation configurations on a chain of length N. It will be useful at this point to separate the state, head location, and work tape (Tracks 4 and 5) from the witness track (Track 6). We will also separate the first m bits of the witness track y from the remaining $N-2-m$ bits, which we refer to as w. We will denote a well-formed standard basis state in C_N as $|v\rangle|y\rangle|w\rangle$, where $|v\rangle$ is a standard basis state for Tracks 4 and 5, and $|y\rangle|w\rangle$ is a standard basis state for Track 6.

According to Lemmas 2.7 and 2.6, the configuration graph defined on the set of well-formed clock configurations consists of paths of relatively short length ($\leq p(N)$) and cycles of the correct clock states. We will first handle the the part of the Hilbert space in which the clock configuration is in one of these short paths. Let $\mathcal{H}_{path}$ be the Hilbert space spanned by the states whose clock configuration is in a path in the configuration graph. $H_{N,prop}$ is the only term in the Hamiltonian that is not diagonal in the standard basis and $H_{N,prop}$ is closed on $\mathcal{H}_{path}$. Therefore, we can lower bound the the eigenvalues of H restricted to $\mathcal{H}_{path}$ separately from the rest of the space.

Lemma 2.12. **[Eliminating Incorrect Clock States]** *The lowest eigenvalue of H_N restricted to $\mathcal{H}_{path}$ is at least $(1-\cos(\pi/(2p(N)+1)))$.*

Proof. Let $|c_0\rangle,\ldots,|c_l\rangle$ be the clock states in a path in the clock configuration graph. According to Lemma 2.6, $l \leq p(N)-1$. Consider a state $|v\rangle|y\rangle|w\rangle$ for the computation tracks from C_N. Starting in state $|c_0\rangle|v\rangle|y\rangle|w\rangle$, after t applications of the transition function, the state of the system will be $|c_t\rangle|v'\rangle|y\rangle|w\rangle$, where $|v'\rangle$ is determined by (c_0,v,y,w,t). We will call this state $|\phi(c_0,v,y,w,t)\rangle$, where $\phi(c_0,v,y,w,t) \in C_N$. Since the operations performed on the computation tracks are always permutations, the following set of states is ortho-normal:

$$S(c_0) = \{|c_t\rangle|\phi(c_0,v,y,w,t)\rangle : \text{ for } |v\rangle|y\rangle|w\rangle \in C_N,\ 0 \leq t \leq l\}.$$

The propagation term $H_{N,prop}$ is block diagonal on the space spanned by this set. Each block is specified by a choice of $|v\rangle|y\rangle|w\rangle$. If we express $H_{N,prop}$ in the standard basis, then each block is $l \times l$ and is the Laplacian of a path of length l.

The set $S(c_0)$ is the only set of states in the basis of $\mathcal{H}_{wf}$ that has clock states $|c_0\rangle,\ldots,|c_l\rangle$. Since the terms $H_{N,init}+H_{N,length}+H_{N,V}$ are diagonal in the standard basis, they are all closed on the span of $S(c_0)$. Since these terms are also positive semi-definite, we need only lower bound the smallest eigenvalue of $H_{N,prop}+H_{N,cl}$ on this space. According to Lemma 2.6, the final clock configuration in the path is illegal, which means that the upper-left element of the matrix will have an additional +1 from $H_{N,cl}$. According to Lemma 2.1, the smallest eigenvalue in this space is at least $(1-\cos(\pi/(2p(N)+1)))$. □

We can now focus on the space of correct clock states. According to Lemma 2.11, if the system starts out in state $|c_{T,s,0}\rangle|\phi\rangle$, where $\phi \in C_N$, then $p(N)$ steps later, the clock configuration will be $|c_{T,s+1,0}\rangle$ and the computation tracks will have returned to state $|\phi\rangle$. Therefore, if we fix the computation state to be some $|v\rangle|y\rangle|w\rangle \in C_N$ at time $(0,0)$, the state of the computation tracks at some time (s,t), depends only on t and not on s or T. We will call this state $|\phi(v,y,w,t)\rangle$. Since the operations performed on the computation tracks are always permutations, the following set of states is orthonormal:

$$\begin{aligned} S_{cyc} = \{|c_{T,s,t}\rangle|\phi(v,y,w,t)\rangle : \text{ for } \\ |v\rangle|y\rangle|w\rangle \in C_N,\ T \in \{0,\ldots,N-3\}, \\ s \in \{0,\ldots,2T\},\ t \in \{0,\ldots,p(N)-1\}\}. \end{aligned}$$

Note that since the verifier V is classical, $|\phi(v,y,w,t)\rangle$ is also a classical standard basis state. The matrix for $H_{N,prop}$ expressed in the basis S is block diagonal. Each block is specified by a value for T and starting configuration $|v\rangle|y\rangle|w\rangle$ for the computation tracks. The block has dimension $(2T+1)p(N)$ and is $1/2$ times the Laplacian of a cycle of length $(2T+1)p(N)$. All other terms are diagonal in this basis.

We will refer to the correct starting configuration for Tracks 4 and 5 as $|v\text{-init}\rangle$. In this correct starting configuration, the contents of Track 5 are ⟨ 1 #* ⟩, the state on Track 4 is q_0, and the head is at the same location as the 1 on Track 5.

There is a correct x for the length N of the chain. This is the string that is written on the computation track after M_{BC} runs for $N-2$ steps starting in $|v\text{-init}\rangle$. The string y is used as the guesses for the responses to the oracle calls. Thus, once x and y are fixed, the inputs to the oracles $x_1,\ldots,x_m$ are determined. In addition, the function $T(x,y)$ is also determined. Since the state of the witness track (including $|y\rangle$) does not change over time, the values of $x_1,\ldots,x_m$ and $T(x,y)$ are well-defined for a block. The verifier V will be run on the oracle queries for the cases in which the string y guesses that the query string is in L. Define the set

$$Y_{rej} = \{y \mid \text{ for some } i \in [m],\ y_i = 1 \text{ and } x_i \notin L\}.$$

In order words, Y_{rej} is the set of oracle guesses for which there is at least one oracle call in which the guess is *yes* but the correct answer is *no*.

We will now partition S_{cyc} into four different sets and consider the subspace spanned by each set separately. The states from S_{cyc} will be put into one of the four sets in blocks. That is, each block is completely contained in one of the four sets.

S_1 consists of the states of the form $|c_{T,s,t}\rangle|\phi(v,y,w,t)\rangle$ in which $|v\rangle \neq |v\text{-init}\rangle$. These are the states from the blocks that do not start in the correct initial configuration.

S_2 consists of the states of the form $|c_{T,s,t}\rangle|\phi(p,y,w,t)\rangle$, where $|v\rangle = |v\text{-init}\rangle$, but $T(x,y)$ is not equal to the length of Track 3 timer.

S_3 consists of the states of the form $|c_{T,s,t}\rangle|\phi(v,y,w,t)\rangle$ where $|v\rangle = |v\text{-init}\rangle$, $T(x,y)$ is equal to the length of the Track 3 timer, and $y \in Y_{rej}$.

S_4 consists of the states of the form $|c_{T,s,t}\rangle|\phi(v,y,w,t)\rangle$ where $|v\rangle = |v\text{-init}\rangle$, $T(x,y)$ is equal to the length of the Track 3 timer, and $y \notin Y_{rej}$.

$H_{N,prop}$ is closed on each S_i because the states from each block are completely contained in S_i or are disjoint from S_i. All of the other Hamiltonian terms besides $H_{N,prop}$ are all diagonal in the standard basis and are therefore diagonal on the basis S_{cyc}.

If $N = N(x)$ for some x, then staring with the correct input configuration $|v\text{-init}\rangle|y\rangle|w\rangle$, if we run the process M_{BC} for $N-2$ steps followed by M_{TV} for $N-2$ steps, the contents of the work track should be $\textsc{Out}(x, yw, M, V)$, the same tape contents as M_{TV} on input (x, yw) when it halts (See Figure 3). Note that the second input parameter to M_{TV} is ω, is the entire binary string on Track 6, which is the concatenation of y and w.

Lemma 2.13. **[Eliminating States from S_1, S_2, and S_3]** *Suppose that the process $(M_{BC})^{N-2}$ followed by $(M_{TV})^{N-2}$ starting in configuration $|v\text{-init}\rangle|y\rangle|w\rangle$ produces $\textsc{Out}(x, yw, M, V)$ on the work track. Then the smallest eigenvalue of H in the span of $S_1 \cup S_2 \cup S_3$, is at least $(1/8)(1-\cos(\pi/(2p(N)+1)))$.*

Proof. Since the terms are all closed on the spans of S_1, S_2, and S_3, we can lower bound each space separately. For a fixed (T, x, y, w), consider the block defined by states of the form $|c_{T,s,t}\rangle|\phi(v,y,w,t)\rangle$. If $|v\rangle \neq |v\text{-init}\rangle$, then the configuration of Tracks 4 and 5 is not equal to $|v\text{-init}\rangle$ at time $(0,0)$. Since the $\circlearrowleft_8$ does not alter Tracks 4 and 5 as it sweeps left, the state of Tracks 4 and 5 will not be equal to $|v\text{-init}\rangle$ for the preceding $N-2$ time steps. There will be at least one point in time when the $\circlearrowleft_5$ is sweeping left that an illegal configuration from $H_{N,init}$ is hit. Call this time t'. Then for all s:

$$H_{N,init}|c_{T,s,t'}\rangle|\phi(v,y,w,t')\rangle = |c_{T,s,t'}\rangle|\phi(v,y,w,t')\rangle$$

Note that there may be more locations where an illegal configuration is reached but we need only consider one as the others can only increase the energy of a state. The diagonal matrix corresponding to $H_{N,init}$ will have a 1 at each location of the form $t' + s \cdot p(N)$, where $s \in \{0,\ldots,2T\}$. If we add this matrix to $H_{N,prop}$ which is $1/2$ times the Laplacian of a cycle graph of length $(2T+1)\cdot p(N)$, then according to Lemma 2.3, the smallest eigenvalue of the sum of the matrices is at least $(1/8)(1-\cos(\pi/(2p(N)+1)))$. This reasoning can be applied to each block in the span of S_1 which gives a lower bound of $(1/8)(1-\cos(\pi/(2p(N)+1)))$ for any eigenvalue in this space.

A similar reasoning can be applied to the span of S_2 using the operator $H_{N,length}$. If a block is inside S_2, then it is using the correct initial configuration $|v\text{-init}\rangle$ which produces the correct input x. Therefore the string y from the witness track and x determine the correct value $T(x,y)$ for the length of the clock on Track 3. This correct value is written in unary on the computation track at the end of the computation M_{TV}, which by assumption of the lemma corresponds to the contents of the work track after $(M_{BC})^{N-2}$ followed by $(M_{TV})^{N-2}$ applied to $|v\text{-init}\rangle|y\rangle|w\rangle$. The contents of the track are $(\sigma_A+\sigma_R)(\sigma_X)^{T(x,y)}$. Since the first σ_A or σ_R symbol counts as a unary digit, the value encoded is $T(x,y)+1$. This is checked against the length of the prefix $\oplus^j \rightarrow \oplus^{T-j}$ or $\oplus^{T-j} \leftarrow \oplus^j$ on Track 3, which is equal to the timer length T plus 1. (The plus 1 comes from the pointer on Track 3.) The Track 1 pointer $\circlearrowleft_1$ will reach an illegal configuration if and only if the Track 3 clock does not have the correct length $T(x,y)$. As with the reasoning for S_1, there may be more than one such violation, but we need only consider one that occurs regularly at times (s,t') for some fixed t' and $s \in \{0,\ldots,2T\}$. Since S_2 consists of those blocks in which the Track 3 timer has the incorrect length, the smallest eigenvalue of $H_{N,prop} + H_{N,length}$ in the span of S_2 will be at least $(1/8)(1-\cos(\pi/(2p(N)+1)))$.

A similar reasoning can be applied to the span of S_3 using the operator $H_{N,V}$. If a block is inside S_3, then it is using the correct initial configuration $|v\text{-init}\rangle$ which produces the correct input x. Therefore the string y from the witness track and x determine the oracle queries $x_1,\ldots,x_m$. If for any $i \in \{1,\ldots,m\}$, $y_i = 1$ and $x_i \notin L$, the verifier will be run on input x_i and will reject, regardless of the string used as witness. By assumption of the lemma, the contents of the work track after the process $(M_{BC})^{N-2}$ followed by $(M_{TV})^{N-2}$ starting in configuration $|v\text{-init}\rangle|y\rangle|w\rangle$ will be the correct output of M_{TV} on input (x, yw). This means that there will be an σ_R symbol written in the first location of the work tape after Segment 4. In this case, Track 1 pointer $\circlearrowleft_1$ will reach an illegal configuration from h_V. Since S_3 consists of those blocks in which $y \in Y_{rej}$, the smallest eigenvalue of $H_{N,prop} + H_{N,V}$ in the span of S_3 will be at least $(1/8)(1-\cos(\pi/(2p(N)+1)))$. □

Finally, we arrive at the space spanned by S_4. Define

$$S_{y,w} = \{|c_{T(x,y),s,t}\rangle|\phi(v\text{-init},y,w,t)\rangle : 0 \le s \le 2T(x,y), 0 \le t < p(N)\}$$

$S_{y,w}$ is the set of standard basis states for a single block, where the initial state is $|v\text{-init}\rangle|y\rangle|w\rangle$ and the Track 3 timer is the correct $T(x,y)$. Define S_y to be the union of $S_{y,w}$ over all possible w. Note that S_4 is the union of all S_y, where $y \notin Y_{rej}$. All of the terms in H are closed on the span of each $S_{y,w}$.

Lemma 2.14. **[Smallest Eigenvalues for Blocks in S_4]** *Suppose that the process $(M_{BC})^{N-2}$ followed by $(M_{TV})^{N-2}$ starting in configuration $|v\text{-init}\rangle|y\rangle|w\rangle$ produces $\textsc{Out}(x, yw, M, V)$ on the work track. If $y \notin Y_{rej}$, then the minimum eigenvalue of the space spanned by S_y is exactly*

$$E(x,y) = \left(1-\cos\left(\frac{\pi}{L+1}\right)\right),$$

where $L = (2T(x,y)+1)\cdot p(N)$.

Proof. All of the terms in H except for $H_{N,prop}$, $H_{N,V}$, and $H_{N,final}$ are 0 on the entire space spanned by S_4 and are therefore 0 on any S_y such that $y \notin Y_{rej}$. Consider $H_{N,prop}$ restricted to the span of a particular $S_{y,w}$ and expressed in the $S_{y,w}$ basis. This matrix is $1/2$ times the Laplacian of a cycle of length $(2T(x,y)+3)\cdot p(N)$. The term $H_{N,final}$ adds a $+1/2$ to two consecutive locations along the diagonal. Since $H_{N,V}$ is semi-positive definite, by Lemma 2.2, the smallest eigenvalue for this block is at least $E(x,y)$.

V is only run on those i such that $y_i = 1$. If $y \notin Y_{rej}$, we know that each such x_i is in L which means that there is a witness which will cause V to accept. Therefore, there is a w such that $H_{N,V}$ applied to every state in $S_{y,w}$ is 0. The lowest eigenvalue of H restricted to the space spanned by this $S_{y,w}$ is exactly the lowest eigenvalue of

$H_{N,prop} + H_{N,final}$ restricted to this space, which by Lemma 2.2, is $E(x, y)$. □

We are finally ready to establish that the smallest eigenvalue occurs within blocks S_y, where y is the set of correct oracle responses, which we refer to as $\tilde{y}$.

Lemma 2.15. **[The Ground Energy Corresponds to Blocks with Correct Oracle Guesses]** *Suppose that the process $(M_{BC})^{N-2}$ followed by $(M_{TV})^{N-2}$ starting in configuration $|v\text{-init}\rangle|y\rangle|w\rangle$ produces $\textsc{Out}(x, yw, M, V)$ on the work track. The smallest eigenvalue for $H_N|_{\mathcal{H}_{br}}$ is $E(x, \tilde{y})$, where $\tilde{y}$ are the correct oracle responses for input x.*

Proof. Recall that throughout the construction, we analyzed the smallest eigenvalue for H_N restricted to the space spanned by all bracketed states.

Note that since $\tilde{y} \notin Y_{rej}$, according to Lemma 2.14, there is an eigenvector whose eigenvalue is $E(x, \tilde{y})$, so we only need to show that every other eigenvalue is at least $E(x, \tilde{y})$.

The smallest eigenvalue of any state that is perpendicular to $\mathcal{H}_{wf}$ is at least 1 from $H_{N,wf-cl}$ and $H_{N,wf-co}$. From Lemma 2.1, the smallest eigenvalue of any state in $\mathcal{H}_{path}$ is at least $1 - \cos(\pi/(2p(N) + 1))$ which will always be larger than $E(x, y)$ for any y. By Lemma 2.13, the smallest eigenvalue of any state in the span of $S_1 \cup S_2 \cup S_3$ is at least $(1/8)(1 - \cos(\pi/(2p(N) + 1)))$ which will always be larger than $E(x, y)$ for any y. Therefore, the smallest eigenvalue for H will correspond to an eigenvector in the span of S_4.

We will prove that if $y \neq \tilde{y}$, then the smallest eigenvalue of H restricted to the span of S_y is greater than $E(x, \tilde{y})$. Suppose that the first k bits of y are the same as the first k bits of $\tilde{y}$, but $y_{k+1} \neq \tilde{y}_{k+1}$. Note that the next oracle query x_{k+1} will be the same for y and $\tilde{y}$.

If $x_{k+1} \notin L$, then $\tilde{y}_{k+1} = 0$. If y matches $\tilde{y}$ in the first k bits and has $y_{k+1} = 1$, then $y \in Y_{rej}$ which means that S_y is contained in S_3 and the smallest eigenvalue of H restricted to this space is larger than $E(x, y)$ for any y.

Now suppose that $x_{k+1} \in L$. Then $\tilde{y}_{k+1} = 1$. Consider a string y that matches $\tilde{y}$ in the first k bits and has $y_{k+1} = 0$. The value of $T(x, y)$ for any such y must be at most:

$$T(x, y) = 2^m + 2^m \left[4^{m+1} + \sum_{j=k+2}^{m} 4^{m-j+1} + \sum_{j=1}^{k} \tilde{y}_j \cdot 4^{m-j+1} \right]$$

Meanwhile, the value of $T(x, \tilde{y})$ will be at least

$$T(x, \tilde{y}) = 2^m \left[4^{m+1} + 4^{m-k} + \sum_{j=1}^{k} \tilde{y}_j \cdot 4^{m-j+1} \right]$$

The lower bound for $T(x, \tilde{y})$ is larger than any $T(x, y)$, where y matches $\tilde{y}$ in the first k bits and does not match $\tilde{y}$ on bit $k+1$. Therefore $E(x, y) > E(x, \tilde{y})$ and the smallest eigenvalue for H restricted to S_y is greater than $E(x, \tilde{y})$. □

2.9.2 Bracketed States and the Final Reduction. Throughout the construction, we have analyzed the Hamiltonian restricted to the space spanned by all bracketed states ($\mathcal{H}_{br}$). These states have the leftmost particle in state $\langle$, right rightmost particle in state $\rangle$ and none of the particles in between in state $\langle$ or $\rangle$. We now add an additional term to h to ensure that the ground state is in $\mathcal{H}_{br}$. We will denote a basis of the d-dimensional particles by $|\langle\rangle$, $|\rangle\rangle$, and $|1\rangle, \ldots, |d-2\rangle$. Let

$$\bar{h} = h + \sum_{j-1}^{d-2} \frac{1}{4} \left(|j\rangle\langle j| \otimes I + I \otimes |j\rangle\langle j| \right) + I \otimes |\langle\rangle\langle\langle\langle| + |\rangle\rangle\langle\rangle\rangle| \otimes I$$

Let $\bar{H}_N$ denote the Hamiltonian resulting from applying $\bar{h}$ to each pair of neighboring particles in a chain of length N.

Lemma 2.16. $\lambda_0(\bar{H}_N) = \lambda_0(H_N|_{\mathcal{H}_{br}}) + \frac{N-2}{2}$

Proof. The first two terms in $\bar{h}$ give a 1/4 penalty to each end particle if it is not in a bracket states and 1/2 penalty to each middle particle if it is not in a bracket state. The next two terms cause a penalty of +1 for any pair with a $\langle$ on the right and a +1 for any pair with a $\rangle$ on the left.

Since $\bar{H}_N$ is closed on $\mathcal{H}_{br}$ and $\mathcal{H}_{br}^{\perp}$, we can consider each space separately. We will first consider the energy penalty from the additional terms in $\bar{h}$. The bracketed states will have an additional energy of exactly $(N-2)/2$ from the additional terms, since they add 1/2 for every particle, except the two at the ends. Therefore, the lowest energy of any state in $\mathcal{H}_{br}$ will be $(N-2)/2 + \lambda_0(H_N|_{\mathcal{H}_{br}}) < (N-2)/2 + 1/4$. Note that the bound in Lemma 2.15, on $\lambda_0(H_N|_{\mathcal{H}_{br}})$ is less than 1/4 for any $N \geq 4$.

Consider a state which is not bracketed. We will only consider the energy from the additional terms which will give a lower bound for the energy for the state. If the particle on the left is not in state $\langle$, then replacing it with $\langle$ will cause the energy to go down by at least 1/4. Similarly, If the particle on the right is not in state $\rangle$, then replacing it with $\rangle$ will cause the energy to go down by at least 1/4. If there is a particle in a bracketed state in the middle, replacing it with a non-bracketed state will cause the energy to go down by at least 1/2. The process can be repeated until the state is bracketed at which point the energy from the additional terms will be exactly $(N-2)/2$. Therefore any non-bracketed configuration will have energy at least $(N-2)/2 + 1/4$. □

We are finally ready to put the pieces together to prove that Function-TIH is hard for FP^{NEXP}.

Theorem 2.17. **[Function-TIH is hard for FP^{NEXP}]** *For any $f \in \text{FP}^{NEXP}$, there is a Hamiltonian h that operates on two d-dimensional particles. Let H_N denote the Hamiltonian on a chain of N d-dimensional particles resulting from applying h to each neighboring pair in the chain. There is a polynomial time computable function $N(x)$ and polynomial $q = 32N^4p(N)^4 = O(N^{12})$ such that, the value of $f(x)$ can be computed in time polynomial in $|x|$, given a value E such $|E - \lambda_0(H_N)| \leq 1/q(N)$.*

Proof. We have shown a construction for h based on the Turing Machine M that computes $f(x)$ and the TM V that is the verifier for the oracle language L in NEXP. Recall that the function $N(x)$ is the number such that after $(N(x)-2)$ TM steps, the binary counter TM completes an increment operation with x on the work tape. Suppose that for some string x, $N = N(x)$ and let m be the number of oracle queries made by M on input x. Suppose further that for every $\omega \in \{0,1\}^{N-4}$, the process $(M_{BC})^{N-2}$ followed by $(M_{TV})^{N-2}$ starting in configuration $|v\text{-init}\rangle|y\rangle|w\rangle$ produces $\textsc{Out}(x, \omega, M, V)$ on the

work track, where $yw = \omega$ and $|y| = m$. Then by Lemma 2.15, the ground energy of $H_N|_{\mathcal{H}_{\mathcal{H}_{br}}}$ is

$$E(x, \tilde{y}) = \left(1 - \cos\left(\frac{\pi}{L+1}\right)\right), \tag{6}$$

where $L = (2T(x, \tilde{y}) + 1) \cdot p(N)$ and $\tilde{y}$ is the string denoting the correct oracle responses on input x. Lemma 2.16 establishes that the state corresponding to the ground energy is in $\mathcal{H}_{br}$.

The Turing Machine M_{TV} ensures that the conditions of Lemma 2.15 are met. Namely, it is in fact the case that for $N = N(x)$ and for every $\omega \in \{0,1\}^{N-4}$, the process $(M_{BC})^{N-2}$ followed by $(M_{TV})^{N-2}$ starting in configuration $|v\text{-init}\rangle|y\rangle|w\rangle$ produces $\textsc{Out}(x, \omega, M, V)$ on the work track.

The propagation terms of the Hamiltonian implement the TM computations on the computation tracks in each iteration of the clock for Tracks 1 and 2 and the ground energy for H_N is equal to the expression given in Equation (6).

Suppose you are given a value E where $|E - E(x, \tilde{y})| < 1/2N^4p(N)^4$. We will show that the value of $f(x, \tilde{y}) = f(x)$ can be computed in polynomial time. If L and L' are distinct positive integers, then

$$\left|\cos\left(\frac{\pi}{L+1}\right) - \cos\left(\frac{\pi}{L'+1}\right)\right| \geq \frac{1}{L^4}.$$

Therefore, the value of L is uniquely determined, given a value E such that $|E - E(x, \tilde{y})| \leq 1/2L^4$, where $L = (2T(x, \tilde{y})+1) \cdot p(N)$. Note that in order for the construction to work, $T(x, y)$ must be written in unary on the work tape, which means that $T(x, y) \leq N - 2$. Therefore, $L \leq 2Np(N)$, and as long as the estimate is within $32N^4p(N)^4$ of the true ground energy, L can be uniquely recovered.

Given E, the value of integer L can be found by binary search in time $O(\log L)$. Since the value of N is known, then $T(x, \tilde{y})$ can be recovered. The low order bits in the binary representation of $T(x, \tilde{y})$ are exactly $f(x, \tilde{y})$, due to the assumption that the length of the output $f(x)$ is at most m, the number of oracle queries, which is a valid assumption due to the padding argument given in the full version [2]. The value of all the numbers in the calculations are polynomial in N and therefore take time polynomial in $\log N$, which is polynomial in $|x|$. □

ACKNOWLEDGEMENTS

We are grateful to the Simons Institute for the Theory of Computing, at whose program on the "The Quantum Wave in Computing" this collaboration began. We would also like to thank the following people for fruitful conversations regarding this work: Ignacio Cirac, Toby Cubitt, James Watson, Johannes Bausch, Sevag Gharibian, and David Perez-Garcia. We are especially grateful to Ignacio Cirac for highlighting the importance of studying quantum computational complexity in the thermodynamical limit.

REFERENCES

[1] Dorit Aharonov, Daniel Gottesman, Sandy Irani, and Julia Kempe. 2009. The Power of Quantum Systems on a Line. *Communications in Mathematical Physics* 287, 1 (Jan 2009), 41–65. https://doi.org/10.1007/s00220-008-0710-3
[2] Dorit Aharonov and Sandy Irani. 2021. Hamiltonian Complexity in the Thermodynamic Limit. https://doi.org/10.48550/ARXIV.2107.06201
[3] Dorit Aharonov and Leo Zhou. 2019. Hamiltonian Sparsification and Gap-Simulation. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA (LIPIcs, Vol. 124)*, Avrim Blum (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2:1–2:21. https://doi.org/10.4230/LIPIcs.ITCS.2019.2
[4] A. Ambainis. 2014. On Physical Problems that are Slightly More Difficult than QMA. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*. 32–43.
[5] Johannes Bausch, Toby Cubitt, and Maris Ozols. 2017. The Complexity of Translationally Invariant Spin Chains with Low Local Dimension. *Annales Henri Poincaré* 18, 11 (Oct 2017), 3449–3513. https://doi.org/10.1007/s00023-017-0609-7
[6] Johannes Bausch, Toby S. Cubitt, Angelo Lucia, and David Perez-Garcia. 2020. Undecidability of the Spectral Gap in One Dimension. *Physical Review X* 10, 3 (Aug 2020). https://doi.org/10.1103/physrevx.10.031038
[7] Johannes Bausch, Toby S. Cubitt, and James D. Watson. 2021. Uncomputability of phase diagrams. *Nature Communications* 12, 1 (Jan 2021). https://doi.org/10.1038/s41467-020-20504-6
[8] Sergey Bravyi and David Gosset. 2015. Gapped and gapless phases of frustration-free spin-1/2 chains. *J. Math. Phys.* 56 (03 2015). https://doi.org/10.1063/1.4922508
[9] Toby Cubitt and Ashley Montanaro. 2016. Complexity Classification of Local Hamiltonian Problems. *SIAM J. Comput.* 45, 2 (2016), 268–316. https://doi.org/10.1137/140998287
[10] Toby S. Cubitt, David Perez-Garcia, and Michael M. Wolf. 2015. Undecidability of the spectral gap. *Nature* 528, 7581 (Dec 2015), 207–211. https://doi.org/10.1038/nature16059
[11] Percy Deift, Mary Beth Ruskai, and Wolfgang Spitzer. 2007. Improved Gap Estimates for Simulating Quantum Circuits by Adiabatic Evolution. *Quantum Inf. Process.* 6, 2 (2007), 121–125. https://doi.org/10.1007/s11128-006-0045-y
[12] Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. 2015. Quantum Hamiltonian Complexity. *Foundations and Trends® in Theoretical Computer Science* 10, 3 (2015), 159–282. https://doi.org/10.1561/0400000066
[13] Sevag Gharibian, Stephen Piddock, and Justin Yirka. 2019. Oracle complexity classes and local measurements on physical Hamiltonians. *arXiv:1909.05981* (2019).
[14] Sevag Gharibian and Justin Yirka. 2019. The complexity of simulating local measurements on quantum systems. *Quantum* 3 (Sept. 2019), 189. https://doi.org/10.22331/q-2019-09-30-189
[15] Daniel Gottesman and Sandy Irani. 2013. The Quantum and Classical Complexity of Translationally Invariant Tiling and Hamiltonian Problems. *Theory of Computing* 9, 2 (2013), 31–116. https://doi.org/10.4086/toc.2013.v009a002
[16] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. 2002. *Classical and Quantum Computation.* American Mathematical Society, USA.
[17] Mark W. Krentel. 1986. The complexity of optimization problems. In *Structure in Complexity Theory*, Alan L. Selman (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 218–218.
[18] R. Oliveira and B. Terhal. 2005. The complexity of quantum spin systems on a two-dimensional square lattice. *arXiv:0504050* (2005).
[19] Christos H. Papadimitriou. 1994. *Computational complexity.* Addison-Wesley.
[20] Raphael Robinson. 1971. Undecidability and Nonperiodicity for the Tilings of the Plane. *Invent. Math.* 12 (1971), 177–209.
[21] James D. Watson, Johannes Bausch, and Sevag Gharibian. 2020. The Complexity of Translationally Invariant Problems beyond Ground State Energies. *arXiv:2012.12717* (2020).
[22] James D. Watson and Toby S. Cubitt. 2021. Computational Complexity of the Ground State Energy Density Problem. *arXiv:2107.05060* (2021).