Hardness and ease of curing the sign problem for two-local qubit Hamiltonians

Klassen, Joel; Marvian, Milad; Piddock, Stephen; Ioannou, Marios; Hen, Itay; Terhal, Barbara M.

DOI
10.1137/19M1287511

Publication date
2020

Document Version
Final published version

Published in
SIAM Journal on Computing

SIAM J. COMPUT.
Vol. 49, No. 6, pp. 1332–1362

HARDNESS AND EASE OF CURING THE SIGN PROBLEM FOR TWO-LOCAL QUBIT HAMILTONIANS*

JOEL KLASSEN†‡, MILAD MARVIAN†§, STEPHEN PIDDOCK¶, MARIOS IOANNOU‖, ITAY HEN†† AND BARBARA M. TERHAL‡

Abstract. We examine the problem of determining whether a multiqubit two-local Hamiltonian can be made stoquastic by single-qubit unitary transformations. We prove that when such a Hamiltonian contains one-local terms, then this task can be NP-hard. This is shown by constructing a class of Hamiltonians for which performing this task is equivalent to deciding 3-SAT. In contrast, we show that when such a Hamiltonian contains no one-local terms then this task is easy; namely, we present an algorithm which decides, in a number of arithmetic operations over $\mathbb{R}$ which is polynomial in the number of qubits, whether the sign problem of the Hamiltonian can be cured by single-qubit rotations.

Key words. quantum, computational complexity, stoquastic, sign problem, quantum Monte Carlo

AMS subject classification. 81-08

DOI. 10.1137/19M1287511

1. Introduction.

The sign problem in quantum physics has long been recognized as one of the main impediments of efficient Monte Carlo simulation of quantum many-body systems [33, 26]. Hamiltonians that do not suffer from the sign problem have recently been given the name "stoquastic" [10], a term which aims to capture the relationship between these Hamiltonians and stochastic processes. Many interesting quantum models such as the transverse field Ising model, the Bose–Hubbard model, and a collection of kinetic particles in a position-dependent potential are stoquastic. However, stoquasticity, as introduced in [10], is a basis-dependent concept. It requires that the Hamiltonian of the physical model in question be real and have nonpositive off-diagonal elements in a given basis. For a many-body local Hamiltonian acting on n qubits, this basis is typically a product basis on which the terms of the Hamiltonian act locally and can be efficiently described. The nonpositivity of the off-diagonal elements of a stoquastic Hamiltonian matrix in a particular basis has important con-

*Received by the editors September 16, 2019; accepted for publication (in revised form) August 20, 2020; published electronically December 17, 2020.

https://doi.org/10.1137/19M1287511

Funding: The work of the first, fourth, and sixth authors was supported by ERC grant EQEC 682726. The work of the second and fifth authors was partially supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via U.S. Army Research Office contract W911NF-17-C-0050. The work of the fifth author was partially supported by the Air Force Research laboratory under agreement FA8750-18-1-0044.

†The first and second authors contributed equally to this work.

‡QuTech, Delft University of Technology, Delft, The Netherlands (J.D.Klassen@tudelft.nl, B.M.Terhal@tudelft.nl)

§Research Laboratory of Electronics, MIT, Cambridge, MA 02139 USA (mmarvian@mit.edu)

¶School of Mathematics, University of Bristol, Bristol, UK (stephen.piddock@bristol.ac.uk)

‖Faculty of Physics, Ludwig Maximilian University of Munich, 80799 Munich, Germany (M.Ioannou@tudelft.nl)

††Information Sciences Institute, University of Southern California, Marina Del Rey, CA 90292 USA (itayhen@isi.edu)

sequences. It guarantees, via the Perron–Frobenius theorem, that there exists a set of orthonormal states, spanning the ground state subspace, whose amplitudes are nonnegative in that basis [11]. In addition, the quantum partition function of a stoquastic Hamiltonian can be expressed as a sum of nonnegative, easily computable weights, which implies that Markov chain Monte Carlo algorithms can be used to perform importance sampling of the quantum configuration space to calculate thermal averages of physical observables, using these weights as (unnormalized) probabilities. For this reason, it is said that stoquastic Hamiltonians do not suffer from the sign problem [11, 10]. However, it is important to note that the absence of a sign problem does not necessarily imply polynomial-time convergence of standard Monte Carlo methods [16, 23, 12].

From a computational complexity perspective, the problem of estimating ground state energies of stoquastic local Hamiltonians is considered easier than for general Hamiltonians [10, 9]. Moreover, in the classification of the complexity of estimating ground state energies of local Hamiltonians, stoquastic Hamiltonians appear as the only intermediate class between classical Hamiltonians and general Hamiltonians [14]. Stoquastic local Hamiltonians are of interest not only in quantum complexity theory. In [11] it was shown that deciding whether a stoquastic Hamiltonian is frustration-free is an MA-complete problem. Recently [1] showed that the gapped version of this question is in NP, linking derandomization of MA to NP to the possibility of gap amplification of stoquastic local Hamiltonians.

Motivation for identifying classes of Hamiltonians that are stoquastic clearly comes from both practical and complexity-theoretic perspectives. Given that stoquasticity is basis-dependent, an interesting question arises: under what circumstances can the sign problem be "cured," as coined in [27], by performing local basis changes? This is the main question explored in this paper.

It is worth noting that the sign problem may be resolved by means other than a local basis transformation. Other methods for generating positive-valued decompositions of the partition function include, e.g., resummation techniques wherein negative-valued weights in the decomposition are grouped together with positive ones to form positive "superweights" that can in turn be treated as probabilities in a quantum Monte Carlo algorithm [35, 13, 19]. Other methods also include applying a constant-depth quantum circuit [32]. These other methods are beyond the scope of this paper.

Naturally, devising techniques for obviating or mitigating the sign problem has been a focus of much research in the quantum Monte Carlo (QMC) community since its inception [31, 25, 35, 13, 19]. In particular, the importance of basis choice has been widely recognized (see, e.g., [8, 18, 17, 30]). Recognizing the key role that stoquastic Hamiltonians play both in computational complexity and in physics, a more general algorithmic approach has recently been launched to determine whether a Hamiltonian can be made stoquastic [27, 24, 5]. In this paper we present an important strengthening of these initial results.

Stoquasticity has also attracted attention from the experimental community. In particular there has been a growing interest in engineering Hamiltonian interactions that are not stoquastic [34, 29]. Some of the reasons for this include: enhancing the performance of quantum annealer protocols for optimization [21, 28, 3], realizing universal adiabatic quantum computers [4, 2, 7], and physically emulating quantum many-body systems [29]. Here too, the question of whether and how local basis changes can cure the sign problem is highly relevant, as experimental quantum advantages hinge on the inability to simulate nonstoquastic interactions on classical computers.

2. Previous work. In what follows, we will refer to Hermitian matrices that are real and have only nonpositive off-diagonal elements as *symmetric Z-matrices* [6].

A no-go result presented recently by some of the authors of this paper states that the problem of determining whether there exists a sign-curing transformation for general local Hamiltonians is NP-hard when one is restricted to applying particular single-qubit transformations to the Hamiltonian [27]. This result can be summarized as follows.

THEOREM 2.1 (see [27]). *Let H be a 3-local n-qubit Hamiltonian, and let* LocalCliffordSignCure *be the problem of determining whether there exist single-qubit Clifford transformations C_u with $\mathsf{C} = \bigotimes_{u=1}^{n} \mathsf{C}_u$ such that $\mathsf{C}H\mathsf{C}^\dagger$ is a symmetric Z-matrix.* LocalCliffordSignCure *is* NP*-hard. Let H be a 6-local n-qubit Hamiltonian, and let* LocalRealRotSignCure *be the problem of determining whether there exist real single-qubit rotations $R_u \in SO(2)$ with $R = \bigotimes_{u=1}^{n} R_u$ such that RHR^T is a symmetric Z-matrix.* LocalRealRotSignCure *is* NP*-hard.*

Remark. When dealing with k-local Hamiltonians with $k > 2$, it is important to note that two distinct notions of stoquasticity have been defined in [10], namely, there exist termwise-stoquastic Hamiltonians and globally stoquastic Hamiltonians. A globally stoquastic Hamiltonian is a symmetric Z-matrix, while a Hamiltonian which is k-local termwise-stoquastic is one which can be decomposed into k-local terms such that each term is a symmetric Z-matrix. A globally stoquastic Hamiltonian need not be termwise-stoquastic, while a termwise-stoquastic Hamiltonian is always globally stoquastic. The results in Theorem 2.1 hold for both definitions. For the two-local Hamiltonians in this paper one can prove [10] that these notions coincide, and hence we do not distinguish between these two definitions in this paper. We provide a proof of this equivalence in Proposition 4.3 for completeness.

It was also recently shown, by other authors of this paper, that for a particularly broad family of two-local Hamiltonians, namely, arbitrary XYZ Heisenberg Hamiltonians, there is an efficient procedure for determining whether the sign problem can be cured by single-qubit unitary transformations.

THEOREM 2.2 (see [24]). *Let $H = \sum_{u,v} H_{uv}$, $u = 1, \ldots, n$, $v = 1, \ldots, n$, be an n-qubit Hamiltonian with $H_{uv} = a_{XX}^{uv} X_u X_v + a_{YY}^{uv} Y_u Y_v + a_{ZZ}^{uv} Z_u Z_v$, where each a_{kk}^{uv} is given with $O(1)$ bits. There is an efficient algorithm, which we call the* XYZ-algorithm*, that runs in time $O(n^3)$ to decide whether there are single-qubit rotations $U_u \in SU(2)$ with $U = \bigotimes_{u=1}^{n} U_u$ such that $UHU^\dagger$ is a symmetric Z-matrix.*

An essential step in the proof of Theorem 2.2 was to show that single-qubit Clifford transformations suffice as basis changes, reducing the problem to an optimization problem over a discrete set of degrees of freedom.

3. Main results. This paper aims to bridge the gap between these two previous results, and to identify the boundary between classes of Hamiltonians for which curing the sign problem by local basis transformations is hard and those for which this problem is easy. The main results of this paper address the following problem.

DEFINITION 3.1 (LocalSignCure). *Given a two-local n-qubit Hamiltonian,* LocalSignCure *is the problem of determining whether there exists a set of single-qubit unitary transformations $U_u \in SU(2)$ with $U = \bigotimes_{u=1}^{n} U_u$ such that $UHU^\dagger = \tilde{H}$ is a symmetric Z-matrix.*

We colloquially refer to such unitary transformation U as a *sign-curing transformation* and say that the sign problem of a Hamiltonian can be *cured* if such a transformation exists.

The main results of this paper are the following two theorems and constitute a strengthening of Theorems 2.1 and 2.2 to two-local Hamiltonians. In section 5 we will prove the following theorem.

THEOREM 3.2. *There exists a family of a two-local n-qubit Hamiltonians for which* LocalSignCure *is* NP*-complete.*

To prove this, we modify the constructions introduced in [27], thereby reducing the locality of Hamiltonians from three-local (in the case of the single-qubit Clifford group) and six-local (in the case of the single-qubit orthogonal group) to two-local. This result demonstrates that LocalSignCure is hard in general. Theorem 3.2 additionally demonstrates that deciding whether a multiqubit two-local Hamiltonian can be sign-cured by single-qubit Clifford transformations is hard. We show in Appendix B that, in the absence of one-local terms, this task is easy. We should stress here that it is not clear that LocalSignCure is a problem in NP for general Hamiltonians. We expand on this point in section 7.

For a relatively broad subclass of two-local Hamiltonians we can, however, show that finding local basis changes is easy.

THEOREM 3.3. *Let H be an* exactly two-local *n-qubit Hamiltonian, meaning a Hamiltonian of the form $H = \sum_{u,v} H_{uv}$ with $H_{uv} = \sum_{k,l\in\{X,Y,Z\}} (\beta_{uv})_{kl} \sigma_k^u \otimes \sigma_l^v$ with σ_k^u a Pauli matrix of type k, acting on qubit u, and $(\beta_{uv})_{kl}$ is given with $O(1)$ bits. There is an efficient algorithm, using $O(n^3)$ arithmetic operations over $\mathbb{R}$, which solves* LocalSignCure *for H.*

This algorithm is presented in section 6. It employs the XYZ-algorithm, referred to in Theorem 2.2, as a subroutine. It is important to note that, just as in the XYZ-algorithm, this algorithm makes no guarantee that H can be cured; it only efficiently *decides* whether or not H can be cured. An important difference between the new algorithm in Theorem 3.3 and the XYZ-algorithm is that the new algorithm requires finding singular value decompositions of matrices specified by $O(1)$ bits, as well as intersections of vector subspaces, while the XYZ-algorithm requires solving a discrete optimization problem. Since we do not address the question of how a finite-precision implementation of these standard linear algebra operations affects the accuracy with which we decide whether the sign problem of H can be cured, we state our theorem in terms of arithmetic operations over $\mathbb{R}$. However, the algorithm is expected to be numerically stable insofar as repeated composition of orthogonal rotations, and intersections of vector spaces, are numerically stable. A rigorous account of the complexity of this problem would require a finite precision formulation. We will not attempt to do this here.

The upshot of these results is that the presence of local fields can change the complexity class of curing the sign problem of two-local Hamiltonians by single-qubit unitaries from P to NP-complete.

4. Preliminaries. For ease of exposition and reference we start by stating the following observation about two-qubit Hamiltonians.

PROPOSITION 4.1. *A two-qubit Hamiltonian $H = \sum_{k,l=I,X,Y,Z} a_{kl} \sigma_k \otimes \sigma_l$ is a symmetric Z-matrix if and only if $a_{IY} = a_{YI} = a_{XY} = a_{YX} = a_{ZY} = a_{YZ} = 0$ (the matrix is real) and $a_{XX} \leq -|a_{YY}|$ and $a_{IX} \leq -|a_{ZX}|$, $a_{XI} \leq -|a_{XZ}|$ (the matrix has nonpositive off-diagonal elements).*

PROPOSITION 4.2 (see [24]). *Consider a two-qubit Hamiltonian*

$$H = \sum_{k,l=I,X,Y,Z} a_{kl}\sigma_k \otimes \sigma_l,$$

where the two-local term can be concisely represented by the 3×3 *matrix*

$$\beta = \begin{pmatrix} a_{XX} & a_{XY} & a_{XZ} \\ a_{YX} & a_{YY} & a_{YZ} \\ a_{ZX} & a_{ZY} & a_{ZZ} \end{pmatrix}.$$

A pair of single-qubit unitary transformations U_1 *and* U_2 *with action* $H \to (U_1 \otimes U_2)H(U_1 \otimes U_2)^\dagger$ *corresponds to a pair of* $SO(3)$ *rotations* O_1, O_2 *acting on the* β*-matrix:* $\beta \to O_1^\mathsf{T}\beta O_2$.

For the curious reader, an example of a Hamiltonian which is not stoquastic under any single-qubit unitary transformations is provided in Appendix A.

It was claimed in [10], without proof, that a two-local Hamiltonian which is globally stoquastic with respect to a basis is also termwise-stoquastic. We include the proof here.

PROPOSITION 4.3 (see [10]). *A two-local Hamiltonian* H *acting on* n *qubits is a symmetric* Z*-matrix in the computational basis if and only if* $H = \sum_{u<v} D_{uv}$, *where each* D_{uv} *acts nontrivially on at most two qubits, namely, qubits* u *and* v, *and* D_{uv} *is a symmetric* Z*-matrix.*

Proof. Let $|x\rangle$, with $x \in \{0,1\}^n$, denote a computational basis state. If there exists a decomposition $H = \sum_{u<v} D_{uv}$ such that D_{uv} is real and $\forall x \neq y$, $\langle x| D_{uv} |y\rangle \leq 0$, then H is real and $\forall x \neq y$, $\langle x| H |y\rangle = \sum_{u,v} \langle x| D_{uv} |y\rangle \leq 0$. This proves one direction of the biconditional; we now prove the other direction. Since H is real, $H = H^\mathsf{T}$. Therefore every Pauli operator P in the Pauli expansion of H must satisfy $P = P^\mathsf{T}$, and so H does not contain any Pauli operators with odd numbers of Y terms. Let $d_H(x,y)$ denote the Hamming distance between bit strings x and y. Since H is two-local, $H = M^{(0)} + M^{(1)} + M^{(2)}$, where $\langle x| M^{(m)} |y\rangle = 0$ whenever $d_H(x,y) \neq m$. In other words the Hamiltonian decomposes into three sets: $M^{(0)}$ contains all terms which are diagonal (i.e., terms of the form ZI, IZ, and ZZ), $M^{(1)}$ contains all terms that flip one bit (i.e., of the form XZ, ZX, XI, and IX), and $M^{(2)}$ contains all terms that flip two bits (of the form XX and YY). There is no particular condition which has to be fulfilled for the diagonal group $M^{(0)}$, and so we ignore it. Furthermore, from the condition $\forall x \neq y$, $\langle x| H |y\rangle \leq 0$, it follows that $\forall x \neq y$, $\langle x| M^{(1)} |y\rangle \leq 0$ and $\langle x| M^{(2)} |y\rangle \leq 0$, since $M^{(1)}$ and $M^{(2)}$ are nonzero at different off-diagonal positions.

For any potential decomposition $H = \sum_{u<v} D_{uv}$ we can similarly write $D_{uv} = D_{uv}^{(0)} + D_{uv}^{(1)} + D_{uv}^{(2)}$, grouping diagonal, one-qubit flipping, and two-qubit flipping terms. Since $M^{(m)}$ contains all terms which flip m-qubits, $M^{(m)} = \sum_{u,v} D_{uv}^{(m)}$. In the case of $m = 2$, $D_{uv}^{(2)}$ and $D_{wr}^{(2)}$ are nonzero at different off-diagonal positions when $u, v \neq w, r$, and so $\forall x \neq y$, $\langle x| M^{(2)} |y\rangle \leq 0$ implies $\forall x \neq y, \forall u < v, \langle x| D_{uv}^{(2)} |y\rangle \leq 0$.

In the case of $m = 1$, $D_{uv}^{(1)}$ and $D_{wx}^{(1)}$ may both be nonzero on the same off-diagonal position, and so we must use a different argument. We can write $M^{(1)} = \sum_{u,v:\, u<v} [a_{XZ}^{uv} X_u Z_v + a_{ZX}^{uv} Z_u X_v] + \sum_u a_X^u X_u$. By writing out matrix elements one can show that

$$\forall x \neq y, \langle x| M^{(1)} |y\rangle \leq 0 \Rightarrow \forall u \;\; a_X^u + \sum_{v:\, v>u} \Delta^v a_{XZ}^{uv} + \sum_{w:\, w<u} \Delta^w a_{ZX}^{wu} \leq 0$$

for all choices of sign-patterns $\Delta^u = \pm 1$. Note that $\Delta^u = \pm 1$ since Z_u is applied on the identical uth bit in x and y, which can be either 0 or 1. This implies that $\forall u$ we have

$$a_X^u \leq -\left(\sum_{v:\, v>u} |a_{XZ}^{uv}| + \sum_{w:\, w<u} |a_{ZX}^{wu}| \right). \tag{4.1}$$

A local term is of the form $D_{uv}^{(1)} = a_{XZ}^{uv} X_u Z_v + a_{ZX}^{uv} Z_u X_v + a_{XI}^{uv} X_u I_v + a_{IX}^{uv} I_u Z_v$, where the coefficients a_{XI}^{uv}, a_{IX}^{uv} can be freely chosen up to the overall constraint $a_X^u = \sum_{v:\, v>u} a_{XI}^{uv} + \sum_{w:\, w<u} a_{IX}^{wu}$. Now, clearly, if (4.1) holds, then one can always distribute a_X^u into a sum over a_{XI}^{uv} (for $v > u$) and a_{IX}^{wu} (for $w < u$) such that each $a_{XI}^{uv} \leq -|a_{XZ}^{uv}|$ and each $a_{IX}^{wu} \leq -|a_{ZX}^{wu}|$. Hence, by Proposition 4.1 there is a decomposition with terms D_{uv} such that $D_{uv}^{(1)}$ is a symmetric Z-matrix, and so D_{uv} is a symmetric Z-matrix. ∎

5. LocalSignCure for a class of two-local Hamiltonians is NP-complete. In this section we present a family of Hamiltonians for which solving LocalSignCure is NP-complete, and thus show that LocalSignCure is NP-hard.

We will first show that LocalSignCure for this class of Hamiltonians is in NP. This is not immediately apparent, since local basis transformations have a continuous parametrization, and hence one either has to allow for approximate sign-curing transformations or prove that for this particular class of Hamiltonians any sign-curing transformation is a member of a discrete subset of transformations. We settle this problem by proving in Lemma 5.1 that with the addition of ancilla qubits and "gadget" interactions, any Hamiltonian in this class can be converted into one for which any sign-curing transformation must consist of either Hadamard gates or the identity operation. In order to prove that the problem is NP-hard, we show how to encode any 3-SAT instance into the problem of curing a corresponding Hamiltonian using the identity or Hadamard gates. In Lemma 5.4 we prove that such a curing transformation exists if and only if the corresponding 3-SAT instance is satisfiable. A proof of Theorem 3.2 follows straightforwardly by considering LocalSignCure for the family of Hamiltonians constructed by adding the gadgets (Lemma 5.1) to the Hamiltonians corresponding to 3-SAT instances (Lemma 5.4).

5.1. Hadamard sign-curing gadget. In this section we introduce the "gadget" interactions which will effectively force any sign-curing transformation to be from a discrete subset of transformations. Let W_u be a single-qubit Hadamard on qubit u; this is a convention we will use throughout this section and the next.

LEMMA 5.1. *Let H be a two-local Hamiltonian on n qubits. For each qubit $u \in \{1, \ldots, n\}$, add three ancilla qubits a_u, b_u, c_u and define the two-local gadget Hamiltonian G and the total Hamiltonian H_{Had} as*

$$\begin{aligned} G = \sum_{u=1}^{n} \big[& -(X_{c_u} + Z_{c_u}) - (X_u X_{a_u} + Y_u Y_{a_u} + Z_u Z_{a_u}) \\ & - (3X_{a_u} X_{b_u} + Y_{a_u} Y_{b_u} + 2Z_{a_u} Z_{b_u}) - (X_{b_u} X_{c_u} + Y_{b_u} Y_{c_u} + Z_{b_u} Z_{c_u}) \big], \end{aligned}$$

$$H_{\mathrm{Had}} = H + G. \tag{5.1}$$

Then the following are equivalent:

1. *There exists a unitary $U = \bigotimes_{u=1}^{n} (U_u \otimes U_{a_u} \otimes U_{b_u} \otimes U_{c_u})$ such that $U H_{\mathrm{Had}} U^\dagger$ is a symmetric Z-matrix.*

2. *There exists* $x \in \{0,1\}^n$ *such that* $\mathsf{W}(x)^\dagger H \mathsf{W}(x)$ *is a symmetric* Z*-matrix, where* $\mathsf{W}(x) = \bigotimes_{u=1}^n W_u^{x_u}$.

Proof. First we prove $2 \to 1$. If there exists $x \in \{0,1\}^n$ such that $\mathsf{W}(x) H \mathsf{W}(x)^\dagger$ is a symmetric Z-matrix, then it is easy to check that $U H_{\mathrm{Had}} U^\dagger$ is a symmetric Z-matrix with

$$U = \bigotimes_{u=1}^n (W_u \otimes W_{a_u} \otimes W_{b_u} \otimes W_{c_u})^{x_u}.$$

To prove the other direction, we will show that if 1 holds, each of the single-qubit unitaries U_α ($\alpha \in \bigcup_{u=1}^n \{u, a_u, b_u, c_u\}$) must be from the discrete set $\{I, W, X, XW\}$. This fact will suffice by the following reasoning. First note that conjugating by local X matrices permutes the off-diagonal matrix entries of the Hamiltonian among themselves [27]. So if $U H_{\mathrm{Had}} U^\dagger$ is a symmetric Z-matrix and $U_\alpha \in \{I, W, X, XW\}$, then $\bar{U} H_{\mathrm{Had}} \bar{U}^\dagger$ is also a symmetric Z-matrix, where $\bar{U} = \bigotimes_\alpha \bar{U}_\alpha$ and

$$\bar{U}_\alpha = \begin{cases} I, & U_\alpha = I \text{ or } X, \\ W, & U_\alpha = W \text{ or } XW, \end{cases}$$

since $U H_{\mathrm{Had}} U^\dagger$ and $\bar{U} H_{\mathrm{Had}} \bar{U}^\dagger$ are related by conjugation by local X matrices. Using the fact that the partial trace of a symmetric Z-matrix is also a symmetric Z-matrix, and noting that by tracing out the ancilla qubits of $\bar{U} H_{\mathrm{Had}} \bar{U}^\dagger$ we get $\bar{U} H \bar{U}^\dagger$, we conclude that if $\bar{U} H_{\mathrm{Had}} \bar{U}^\dagger$ is a symmetric Z-matrix, then so is $\bar{U} H \bar{U}^\dagger$, and $\bar{U} = \mathsf{W}(x)$ for some x.

We now proceed with proving that $U_\alpha \in \{I, W, X, XW\}$ given item 1 of Lemma 5.1. Here we make use of the picture of orthogonal rotations on β matrices, as mentioned in Proposition 4.2. For a given u we note that there are no one-local terms involving qubits a_u and b_u, and that the matrix $\beta_{a_u b_u}$ is diagonal and has 3 distinct nonzero singular values. In the absence of one-local terms, it follows directly from Proposition 4.1 that $\beta_{a_u b_u}$ has to remain diagonal for H_{Had} to be a symmetric Z-matrix. Therefore, the only possible transformations are signed permutations (of the Paulis) on qubits a_u and b_u with the permutations being the same to maintain the diagonality of $\beta_{a_u b_u}$. This implies that there exists a single-qubit Clifford transformation C (corresponding to the permutation) and Pauli matrices P_{a_u} and P_{b_u} such that $U_{a_u} = P_{a_u} \mathsf{C}$ and $U_{b_u} = P_{b_u} \mathsf{C}$.

We now consider the interaction between qubits u and a_u. For the overall Hamiltonian to be real, the coefficients of $X_u Y_{a_u}, Y_u X_{a_u}, Z_u Y_{a_u}, Y_u Z_{a_u}$ must all be zero. Since there are no one-local terms acting on qubit a_u, the coefficient of $Z_u X_{a_u}$ must also be zero and so the rotated matrix $\beta'_{u a_u}$ must have zeros in the following positions:

$$\beta'_{u a_u} = O_u^{\mathsf{T}} \beta_{u a_u} O_{a_u} = \begin{pmatrix} * & 0 & * \\ 0 & * & 0 \\ 0 & 0 & * \end{pmatrix}.$$

Note that $-\beta_{u a_u}$ is the identity matrix in (5.1), so $\beta'_{u a_u} = O_u^{\mathsf{T}} \beta_{u a_u} O_{a_u} = -O_u^{\mathsf{T}} O_{a_u}$ is an orthogonal matrix. The only orthogonal matrix with zeros in these positions is a diagonal matrix (with ± 1 on the diagonal). Therefore O_u must equal O_{a_u} up to signs; that is, $U_u = P U_{a_u}$ for some Pauli P.

Since the matrix $\beta_{b_u c_u}$ is identical to $\beta_{u a_u}$ and also there are no one-local terms acting on qubit b_u, an identical argument shows that any curing transformation U_{b_u} and U_{c_u} must satisfy $U_{b_u} = P U_{c_u}$ for some Pauli matrix P. Thus $\forall \alpha \in \{u, a_u, b_u, c_u\}$,

we have $U_\alpha = P_\alpha \mathsf{C}$ for some Pauli matrix P_α and a single-qubit Clifford transformation C.

Due to the one-local terms $-(X_{c_u} + Z_{c_u})$, if C maps $X \to Y$ or $Z \to Y$, the Hamiltonian will have imaginary matrix entries and so, up to multiplication by a Pauli, C must be I or W. Incorporating any such Pauli into P_{c_u}, we may assume w.l.o.g. that $\mathsf{C} \in \{I, W\}$. Furthermore, if P_{c_u} is Y or Z, there will be a positive $+X_{c_u}$ term, so $P_{c_u} \in \{I, X\}$. Finally, if any of the other P_α are Y or Z, there will be a positive $+X \otimes X$ term, and so $\forall \alpha$ we must have $P_\alpha \in \{I, X\}$, and so $U_\alpha = P_\alpha \mathsf{C} \in \{I, W, X, XW\}$. ∎

The following Lemma was proved in [24, 27] by formulating an efficient strategy which finds a two-local termwise-stoquastic decomposition which is equivalent to H being a symmetric Z-matrix by Proposition 4.3.[1]

LEMMA 5.2 (see [24]). *Given a two-local Hamiltonian H on n qubits, one can decide whether H is a symmetric Z-matrix in the given basis in a number of steps polynomial in n.*

Corollary 5.3 now follows immediately from Lemmas 5.1 and 5.2, because the string $(x_1, \ldots, x_n)$ is an efficiently checkable witness in the case that H_{Had} is sign-curable by a local unitary transformation.

COROLLARY 5.3. *If H is a two-local Hamiltonian, then for Hamiltonians H_{Had} of the form in* (5.1)*,* LocalSignCure *is in NP.*

5.2. LocalSignCure is NP-hard. Now we will show how to reduce 3-SAT to LocalSignCure and hence show that LocalSignCure is NP-hard. At the heart of the construction is a Hamiltonian H_{OR} which acts on four qubits labeled $d, 1, 2, 3$:

$$H_{\mathrm{OR}} = -(X_d + Z_d + I) \otimes (Z_1 + Z_2 + Z_3 + 2I). \tag{5.2}$$

Thanks to Lemma 5.1 it suffices to consider a local basis change of the form $\mathsf{W}(x) = \bigotimes_{j \in \{d,1,2,3\}} W_j^{x_j}$. Note that $-(X_d + Z_d + I)$ has nonpositive matrix entries and is invariant under conjugation by W_d. Therefore $\mathsf{W}(x) H_{\mathrm{OR}} \mathsf{W}(x)^\dagger$ is a symmetric Z-matrix if and only if the bit string x is such that all the matrix entries of

$$W_1^{x_1} Z_1 W_1^{x_1} + W_2^{x_2} Z_2 W_2^{x_2} + W_3^{x_3} Z_3 W_3^{x_3} + 2I$$

are nonnegative. Recalling that $WZW = X$, one can see that for any x, all the off-diagonal matrix entries are nonnegative. In addition, the diagonal entries are nonnegative unless $(x_1, x_2, x_3) = (0, 0, 0)$. Therefore $\mathsf{W}(x) H_{\mathrm{OR}} \mathsf{W}(x)^\dagger$ is a symmetric Z-matrix if and only if $x_1 \vee x_2 \vee x_3$ evaluates to true.

Let C be a 3-SAT Boolean formula of the form

$$C = \bigwedge_{k=1}^{m} C_k = \bigwedge_{k=1}^{m} (c_{k,1} \vee c_{k,2} \vee c_{k,3}),$$

with m clauses and n variables, where each $c_{k,j}$ is equal to x_i or $\bar{x}_i$ for some $i \in \{1, \ldots, n\}$.

[1] More generally, one can note that it is easy to decide whether a k-local Hamiltonian is k-local termwise-stoquastic, as this is a linear programming problem. This can be seen by noting that the number of parameters needed to specify a local decomposition is polynomially dependent on the number of qubits, and the number of conditions to test on each term is dependent on the locality of the term. Furthermore, all of the conditions are linear [27, 22].

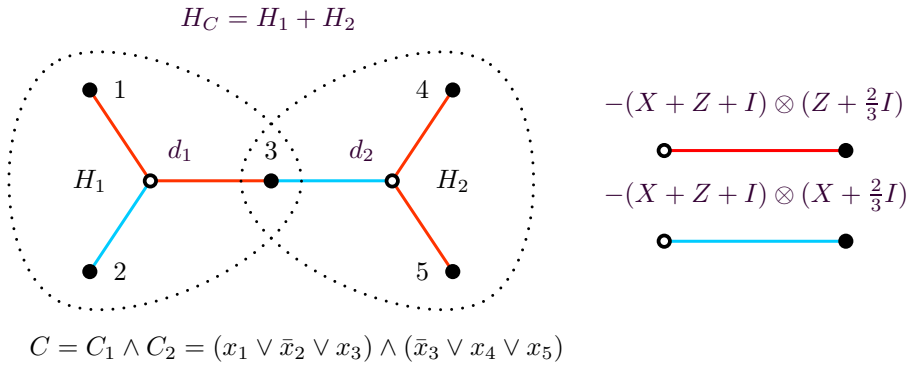

FIG. 1. *An encoding of a 3-SAT Boolean formula C, with two clauses and five variables, into a Hamiltonian H_C as prescribed by* (5.3).

Let H_C be the Hamiltonian on $m+n$ qubits (labeled $\{1,\dots,n\} \cup \{d_1,\dots,d_m\}$) defined by

$$H_C = \sum_{k=1}^{m} H_k = \sum_{k=1}^{m} -(X_{d_k} + Z_{d_k} + I) \otimes (S(c_{k,1}) + S(c_{k,2}) + S(c_{k,3}) + 2I)\,, \tag{5.3}$$

where

$$S(c) = \begin{cases} Z_i & \text{if } c = x_i \text{ for some } i, \\ X_i & \text{if } c = \overline{x_i} \text{ for some } i. \end{cases}$$

An instance of such a Hamiltonian is illustrated in Figure 1. For $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$, define

$$\mathsf{W}(x,y) = \left(\bigotimes_{i=1}^{n} W_i^{x_i}\right) \otimes \left(\bigotimes_{j=1}^{m} W_{d_j}^{y_j}\right).$$

LEMMA 5.4. *Let C be a 3-SAT Boolean formula, let H_C be the corresponding Hamiltonian defined in* (5.3)*, and let $x \in \{0,1\}^n$. $C(x)$ evaluates to true if and only if $\forall y \in \{0,1\}^m$, $\mathsf{W}(x,y) H_C \mathsf{W}(x,y)^\dagger$ is a symmetric Z-matrix.*

Proof. Note that $(X_{a_k} + Z_{a_k} + I)$ is invariant under conjugation by W_{d_k}, so the choice of y leaves H_C unchanged. Furthermore $(X_{a_k} + Z_{a_k} + I)$ has nonnegative matrix entries (with some positive off-diagonal matrix entries). Therefore $\mathsf{W}(x,y) H_k \mathsf{W}(x,y)^\dagger$ is a symmetric Z-matrix if and only if all the matrix entries of

$$\mathsf{W}(x,y)\big(S(c_{k,1}) + S(c_{k,2}) + S(c_{k,3}) + 2I\big)\mathsf{W}(x,y)^\dagger \tag{5.4}$$

are nonnegative. As discussed above, $S(c)$ has been defined so that the matrix entries of (5.4) are nonnegative exactly when $(c_{k,1} \vee c_{k,2} \vee c_{k,3})$ is true.

Since each H_k is the only interaction acting on qubit d_k, and H_k can only fail to be a symmetric Z-matrix due to terms which act nontrivially on d_k, it follows that $\mathsf{W}(x,y) H_k \mathsf{W}(x,y)^\dagger$ must be a symmetric Z-matrix for all k in order for $\mathsf{W}(x,y) H_C \mathsf{W}(x,y)^\dagger$ to be a symmetric Z-matrix. Since $C = \bigwedge_{k=1}^{m} C_k$, this happens exactly when $C(x)$ is true. ∎

This leads to the main result of this section.

COROLLARY 5.5. *There exists a class of two-local Hamiltonians for which* LocalSignCure *is NP-complete.*

Proof. For any 3-SAT formula C, we construct the two-local Hamiltonian $H_{C,\mathrm{Had}}$ by adding the gadget interactions G of (5.1) for each qubit in the Hamiltonian H_C in (5.3). Using Lemmas 5.1 and 5.4, we conclude that satisfying a family of 3-SAT formulae C is equivalent to LocalSignCure for the corresponding family of $H_{C,\mathrm{Had}}$ Hamiltonians, from which we conclude that LocalSignCure is NP-hard. The inclusion of LocalSignCure for $H_{C,\mathrm{Had}}$ in NP follows from Corollary 5.3. ∎

Let us briefly comment on the question of how hard determining the ground state energy of $H_{C,\mathrm{Had}}$ may be. We observe that the qubits d_i and the triples of ancilla qubits a_u, b_u, c_u for each u only couple to the n qubits on which the clauses act. In particular, if we would fix the state of these ancillary qubits to ψ, then the resulting Hamiltonian $\langle\psi| H_{C,\mathrm{Had}} |\psi\rangle$ acting only on the clause qubits would be purely one-local. It is, however, not a priori clear that the minimal energy is obtained when the state ψ is a product state; if this were the case, then the ground state energy problem would be in NP as a prover could provide a description of this product state. However, even the problem of finding such a product ground state is not guaranteed to have an obvious polynomial-time classical algorithm. It would be worthwhile to investigate this further.

6. An efficient algorithm for LocalSignCure for exactly two-local Hamiltonians.

6.1. Preliminaries. In this section we prove Theorem 3.3 by presenting an efficient algorithm for solving LocalSignCure when H is an exactly two-local Hamiltonian.

We represent an exactly two-local Hamiltonian by a graph G with matrix-weighted edges. Each qubit in the Hamiltonian corresponds to a vertex in the graph, and each edge corresponds to a term $H_{uv} \neq 0$. Every edge is weighted by the 3×3 real matrix β_{uv} associated with H_{uv}, as discussed in Proposition 4.2.

In this picture, LocalSignCure reduces to the following problem. Consider a graph $G = (V, E)$ with n vertices in V and a set of directed matrix-weighted edges E. Each edge (u, v) with direction $u \to v$ is weighted by a 3×3 real matrix β_{uv}, and we define $\beta_{vu} = \beta_{uv}^{\mathsf{T}}$.[2] Given G, find a set of $SO(3)$ rotations $\{O_u\}_{u=1}^n$ which have the action $O_u^{\mathsf{T}} \beta_{uv} O_v = \Sigma_{uv}\ \forall \beta_{uv}$, such that for all edges (u, v),

$$\Sigma_{uv} \text{ is a diagonal matrix,} \tag{6.1}$$

$$|(\Sigma_{uv})_{11}| \geq |(\Sigma_{uv})_{22}| \quad \forall \beta_{uv}, \tag{6.2}$$

$$(\Sigma_{uv})_{11} \leq 0 \quad \forall \beta_{uv}. \tag{6.3}$$

Otherwise prove that no such set exists.

Note that we have rephrased the conditions in Proposition 4.1 according to the labeling $X \to 1$, $Y \to 2$, $Z \to 3$. One can argue (see [24]) that if there exist $O(3)$ rotations that perform this task, then one can easily construct a set of $SO(3)$ rotations that do the same. Therefore any orthogonal rotations will suffice.

[2] The purpose of the direction is merely to allow the matrix weight to be well defined. Throughout the text we will ignore the directedness of the graph and treat the edge as though it is weighted by β_{uv} or β_{vu}, depending on our purpose.

If all matrices β_{uv} are diagonal, then the XYZ-algorithm in Theorem 2.2 can be applied. Naively, our problem could then be reduced to the question, Is there a set of rotations $\{O_u\}$ that has the action $O_u^{\mathsf{T}}\beta_{uv}O_v = \Sigma_{uv}\ \forall\beta_{uv}$, such that condition (6.1) is satisfied, and what are those rotations? If this problem is efficiently solved, one may incorporate the algorithm for finding the set of rotations as a subroutine of the XYZ-algorithm and solve the entire problem. However, we show in Appendix D that deciding the existence of such a set of rotations on β_{uv} such that condition (6.1) is satisfied is an NP-hard problem.

Thus a different approach must be taken; namely, we focus on condition (6.2) in order to prune the set of solutions which needs to be considered. More concretely, we will present an algorithm which solves the following problem.

PROBLEM 1 (No-Lone-YY and diagonal). *Is there a set of orthogonal rotations* $\{O_u \in O(3)\}$ *that have the action* $O_u^{\mathsf{T}}\beta_{uv}O_v = \Sigma_{uv}\ \forall\beta_{uv}$*, such that*

$$\Sigma_{uv} \text{ is a diagonal matrix}, \tag{6.4}$$

$$(\Sigma_{uv})_{22} = 0, \quad \forall\beta_{uv} \text{ for which } \mathrm{Rank}\,(\beta_{uv}) = 1? \tag{6.5}$$

If yes, what is that set? Note here that condition (6.4) *is identical to condition* (6.1)*, and condition* (6.5) *is precisely condition* (6.2) *restricted to rank-*1 *matrices.*

Note that an efficient algorithm for this problem can be incorporated into the XYZ-algorithm to produce an efficient algorithm for LocalSignCure for exactly two-local Hamiltonians, thus directly proving Theorem 3.3. More precisely, a solution to this problem prescribes a transformation of our Hamiltonian into an XYZ-Heisenberg Hamiltonian, in which case the XYZ-algorithm can be used to decide if the Hamiltonian can be rotated into a symmetric Z-matrix by single-qubit unitary transformations. Furthermore, if no solution exists to this problem, then rotating the Hamiltonian into a symmetric Z-matrix by single-qubit transformations is impossible, since both of the above conditions are necessary conditions.

An orthogonal transformation O_u can be written as $O_u = (e_u^1, e_u^2, e_3^u)$, where the e_u^i are three real orthonormal column vectors. We can thus view selecting O_u as selecting a basis $b_u = (e_u^1, e_u^2, e_u^3)$ at vertex u.

DEFINITION 6.1 (No-Lone-YY basis). *Given a matrix-weighted graph* G *with weights* β_{uv}*, an ordered assignment of basis vectors* $b_u = (e_1^u, e_2^u, e_3^u)$ *to each vertex in the graph is called a* No-Lone-YY basis (NLY basis) $B = \{b_u\}$*, when* e_i^v *is a right singular vector of* β_{uv} *with corresponding left singular vector equal to* $\pm e_i^u$*, i.e.,*

$$\forall u, v, i:\ \beta_{uv}e_i^v = \pm\sigma e_i^u,\ \ \beta_{uv}^{\mathsf{T}}e_i^u = \pm\sigma e_i^v, \tag{6.6}$$

*and for all rank-*1 *matrices* β_{uv}*,*

$$\beta_{uv}e_2^v = 0,\ \ \beta_{uv}^{\mathsf{T}}e_2^u = 0. \tag{6.7}$$

It is not hard to see that solving Problem 1 is equivalent to finding an NLY basis, or showing that none exists.

It is important to note that if we flip the signs on our basis elements, this will have no bearing on the problem. We formally define this equivalence under sign flips as follows.

DEFINITION 6.2. *Two ordered bases b_u and b'_u are equivalent modulo signs:*

$$b_u = b'_u \text{ modulo signs}$$

if $b_u = (e_1^u, e_2^u, e_3^u)$, and $b'_u = (\delta_1^u e_1^u, \delta_2^u e_2^u, \delta_3^u e_3^u)$ with $\delta_i^u \in \{+1, -1\}$.

Thus throughout the text we will often talk about a basis *modulo signs*, meaning a basis choice where the signs have not been specified. The premise is that the choice of signs is irrelevant for the purposes of the problem. This will prove to be a useful fact in the proofs of Lemma 6.13 and Theorem 6.14.

A final comment on notation. In the next two subsections we will make use of *sets of subspaces* of $\mathbb{R}^3$. We wish to hold onto the notion that these are sets of subspaces, but make use of natural set notation in terms of the elements of the subspaces. Consequently, for ease of exposition, we will abuse notation in the following ways. We denote a set of subspaces by $\mathbb{S} = \{S_i | S_i \subseteq \mathbb{R}^3\}$. We denote the entrywise intersection of sets of subspaces by

$$\mathbb{S}_1 \cap \mathbb{S}_2 := \{S_i \cap S_j | S_i \in \mathbb{S}_1 \, , S_j \in \mathbb{S}_2\}.$$

We denote the span of the union of the subspaces by

$$\text{span}(\mathbb{S}) := \text{span}\left(\bigcup_{S_i \in \mathbb{S}} S_i\right).$$

We say a set of vectors $b = \{\nu | \nu \in \mathbb{R}^3\}$ is in a set of subspaces $\mathbb{S}$, with the notation $b \subseteq \mathbb{S}$, if every vector in b belongs to a subspace in $\mathbb{S}$. Furthermore, we say a set of subspaces $\mathbb{S}_1$ is contained in another set of subspaces $\mathbb{S}_2$, with the notation $\mathbb{S}_1 \subseteq \mathbb{S}_2$, if every subspace in $\mathbb{S}_1$ is contained in a subspace in $\mathbb{S}_2$. The reason these two notations coincide is because it can be helpful for our purposes to conceptualize the vectors in b as one-dimensional subspaces, since we do not care about the sign of the vector. We denote the transformation on each of the subspaces by an orthogonal rotation O as

$$O\mathbb{S} := \{OS_i | S_i \in \mathbb{S}\}.$$

6.2. XOR-SAT. In the next two subsections we will make repeated use of a subroutine for solving the 2-XOR-SAT problem. XOR-SAT is a Boolean satisfiability problem in which one has a set of Boolean variables $\{x_u\}$ and a set of clauses consisting of not operations and xor operations, e.g., $\bar{x}_u \oplus x_v$, and one asks if there exists an assignment to the Boolean variables which satisfies all of the clauses. XOR-SAT is known to be solvable in polynomial time. 2-XOR-SAT is quite trivially solvable in time $O(N^2)$, where N is the number of variables: the assignment of one variable in the clause uniquely determines the assignment of the other variable in the clause. Thus one varies the assignment of one variable, and propagates that choice through the clauses (of which there are worst case N^2), until all variables are assigned or a contradiction is found (if there are disconnected sets of variables, one does the same thing for each disconnected cluster).

6.3. Illustrative subcase: Graphs with rank-1 edges. We begin by considering an illustrative subcase in which each edge in the graph is weighted by a rank-1 matrix (i.e., a *rank-1 edge*). The significance of rank-1 edges is that their matrix weights have a two-dimensional null space, which implies an additional freedom in the choice of basis that is not present in edges weighted by rank > 1 matrices (i.e.,

rank > 1 *edges*), which have at most a one-dimensional null space. This difference will become more apparent when we consider the general case of a graph with both rank > 1 and rank-1 edges.

For a graph with only rank-1 edges the algorithm for solving Problem 1 breaks up into two parts. In the first part we impose some of the necessary constraints for the basis assignment to be NLY, formulating a candidate basis B. In the second part we permute the vectors of the candidate basis so that it could become an NLY basis.

DEFINITION 6.3 (candidate basis of a rank-1 graph). *A* candidate basis of a rank-1 graph *is a basis assignment* $B = \{b_u\}$ *such that for every edge* $e = (u, v)$*, the basis vectors* $b_u = (e_1^u, e_2^u, e_3^u)$ *are eigenvectors of* $\beta_{uv}\beta_{uv}^\mathsf{T}$ *and the basis vectors* $b_v = (e_1^v, e_2^v, e_3^v)$ *are eigenvectors of* $\beta_{uv}^\mathsf{T}\beta_{uv}$.

PROPOSITION 6.4. *Given a rank-*1 *matrix* β_{uv}*, if the basis vectors* b_u *are eigenvectors of* $\beta_{uv}\beta_{uv}^\mathsf{T}$ *and the basis vectors* b_v *are eigenvectors of* $\beta_{uv}^\mathsf{T}\beta_{uv}$*, then there exists a single index* i *such that* $\beta_{uv}^\mathsf{T} e_i^u \neq 0$ *and a single index* j *such that* $\beta_{uv} e_j^v \neq 0$*. Furthermore,* $\exists \sigma \neq 0$ *such that* $\beta_{uv} e_i^v = \pm\sigma e_j^u$ *and* $\beta_{uv}^\mathsf{T} e_j^u = \pm\sigma e_i^v$.

Proof. Since β_{uv} is rank-1 it follows that $\beta_{uv}\beta_{uv}^\mathsf{T}$ and $\beta_{uv}^\mathsf{T}\beta_{uv}$ are also rank-1. Thus only single basis vectors $e_i^u \in b_u$ and $e_j^v \in b_v$ will be eigenvectors with nonzero eigenvalue of $\beta_{uv}\beta_{uv}^\mathsf{T}$ and $\beta_{uv}^\mathsf{T}\beta_{uv}$, respectively. Therefore e_i^u and e_j^v are the only singular vectors in b_u and b_v which have nonzero singular values for β_{uv}. Since the column and row spaces of β_{uv} are both one-dimensional, it must be the case that $\beta_{uv} e_i^v = \pm\sigma e_j^u$ and $\beta_{uv}^\mathsf{T} e_j^u = \pm\sigma e_i^v$ for some $\sigma \neq 0$. ∎

Note that given a candidate basis (and the corresponding orthogonal rotations $\{O_u\}$) the matrix $O_u^\mathsf{T}\beta_{uv}O_v$ has exactly one nonzero entry but isn't necessarily diagonal. An example of a matrix of this form would be

$$O_u^\mathsf{T}\beta_{uv}O_v = \begin{bmatrix} 0 & 0 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{6.8}$$

Therefore a candidate basis is close to being an NLY basis, except the ordering of the basis vectors in b_u and b_v may not be correct. In order to remedy this, we need to permute orderings of the various b_u. To help visualize this, we may consider the edge (u, v) to be labeled i on the u side, and j on the v side, where i and j are the indices specified in Proposition 6.4. For example, the matrix in (6.8) would correspond to the edge in Figure 2. In this picture the candidate basis B thus specifies a bilabeled graph, i.e., a graph where every edge has two labels (denoted by different colors in the article's electronic version, and by solid and broken lines in the print version).

FIG. 2. *Bilabeling of a rank-*1 *edge.*

DEFINITION 6.5 (basis permutation). *Given a basis* $b = (e_1, e_2, e_3)$ *and permutation* π*, the* permuted basis b_u^π *is defined as* $b_u^\pi := (e^u_{\pi^{-1}(1)}, e^u_{\pi^{-1}(2)}, e^u_{\pi^{-1}(3)})$*. Given a basis assignment to every vertex* $B = \{b_u\}$ *and an assignment of permutations to every vertex* $\Pi = \{\pi_u\}$*, the permuted basis assignment is defined as* $B^\Pi := \{b_u^{\pi_u}\}$.

Given that the candidate basis B specifies a bilabeled graph, we can think of the action of basis permutations $b_u \to b_u^\pi$ as a transformation on the labeling, $i \to \pi(i)$, of

every label adjacent to u, as illustrated in Figure 3. The premise is then that the only

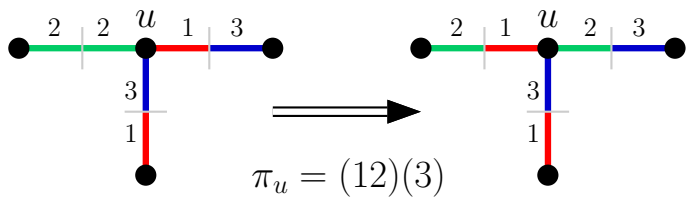

FIG. 3. *Action of permutations on a bilabeled graph.*

remaining task is to find a set of permutations $\{\pi_u \in S_3\}$ to apply to every vertex so that

- the bilabeling is uniform on an individual edge (i.e., $i = j$), corresponding to condition (6.6);
- no edge is labeled by the (green/dashed) value 2, corresponding to condition (6.7).

If we are unsuccessful in finding either a candidate basis B or an appropriate permutation Π, then we will argue that no NLY basis exists.

Algorithm 1: Algorithm for finding a candidate basis of a rank-1 graph

Input : Graph $G = (V, E)$, rank-1 matrix edge weights $\{\beta_{uv}\}$.
Output: A candidate basis $B = \{b_u\}$, if one exists. Otherwise False, indicating no candidate basis exists.

```
for v ∈ V do
    S_v = {R^3}
    for u ∈ V such that e = (u, v) ∈ E do
        S_v^e = the set of orthogonal maximal eigenspaces associated with every
          eigenvalue of β_uv^T β_uv
        S_v = S_v ∩ S_v^e
    if span(S_v) ≠ R^3 then
        return False;
    Choose orthonormal basis b_v = (e_1^v, e_2^v, e_3^v) ⊆ S_v.
return B = {b_v}
```

LEMMA 6.6. *Algorithm* 1 *efficiently finds a candidate basis for a rank-*1 *graph, or otherwise shows that no such candidate basis exists.*

Proof. Any vectors we choose from $\mathbb{S}_v$ must simultaneously be eigenvectors of $\beta_{uv}^{\mathsf{T}}\beta_{uv}$ for all edges $e = (u, v)$ adjacent to v, since they must simultaneously belong to every $\mathbb{S}_v^e$. Furthermore, the spaces in $\mathbb{S}_v$ contain all vectors that are simultaneously eigenvectors of $\beta_{uv}^{\mathsf{T}}\beta_{uv}$ for all edges $e = (u, v)$ adjacent to v. Therefore if $\mathbb{S}_v$ does not span $\mathbb{R}^3$, then we cannot possibly choose a set of orthonormal vectors b_v which are simultaneous eigenvectors of all neighboring edges.

The number of elements in any set of eigenspaces $\mathbb{S}_v^e$ is upper bounded by 3, corresponding to 3 orthogonal one-dimensional subspaces. The same is true for any intersection of any number of these sets. Thus computing any intersection between these sets of subspaces takes $O(1)$ time. Therefore one may iteratively construct $\mathbb{S}_v$ in time proportional to the number of edges. Thus the algorithm is efficient. ∎

We now describe in words the algorithm (Algorithm 2) for finding permutations Π such that B^Π is an NLY basis. This algorithm takes a candidate basis B of a rank-1 graph and finds a set of permutations Π such that B^Π is an NLY basis, or otherwise indicates that no such set of permutations exists.

For each edge (u, v), identify the left singular vector $e_i^u \in b_u$ and corresponding right singular vector $e_j^v \in b_v$ which are not in the null space of β_{uv}, which must exist by Proposition 6.4. Label each rank-1 edge (u, v) with an ordered pair of labels (i, j), as illustrated in Figure 2. We say that an edge $e = (u, v)$ with labeling (i, j) connects to u with label i and connects to v with j.

If for any vertex v there are at least three edges, each connected to v by a different label, then terminate and indicate that the desired set of permutations does not exist.

If the algorithm has not terminated, then for every vertex v there exist two labels i and j such that every edge adjacent to v connects to v with one of those two labels. This holds even if every edge connects to v with the same label. Identify a pair of permutations π_v^0 and π_v^1 such that $\pi_v^0(i) = 1$, $\pi_v^0(j) = 3$ and $\pi_v^1(i) = 3$, $\pi_v^1(j) = 1$.

The task now becomes assigning a binary value x_v to each v so that for every edge (u, v) with label (i, j) the binary assignments satisfy

$$\pi_u^{x_u}(i) = \pi_v^{x_v}(j).$$

By virtue of π never mapping any label to the value 2, and ensuring the uniform bilabeling of each edge, such an assignment will specify an NLY basis.

This problem reduces straightforwardly to an XOR-SAT problem. Each edge (u, v) corresponds to an XOR clause: $\bar{x}_u \oplus x_v$ if $i = j$, and $x_u \oplus x_v$ if $i \neq j$. If there is a solution, then this specifies an NLY basis, namely, B^Π with $\Pi = \{\pi_u^{x_u}\}$. If there is no solution, then the desired set of permutations does not exist.

THEOREM 6.7. *Given a graph with only rank-1 edges, one can efficiently find an NLY basis, or otherwise show that no such basis exists.*

Proof. The algorithm for finding an NLY basis in this case proceeds by first finding a candidate basis B using Algorithm 1, and then finding a set of permutations Π such that B^Π is an NLY basis using Algorithm 2. It should be clear that the basis B^Π is an NLY basis, since for every edge (u, v) Algorithm 2 has explicitly paired those two vectors in b_u and b_v not in the null space of β_{uv}, and ensured that they are not the second entry. Additionally, Algorithm 2 is efficient, since solving 2-XOR-SAT is efficient.

If Algorithm 1 fails, then by Lemma 6.6 no candidate basis exists, and since any NLY basis must satisfy the conditions of being a candidate basis, no NLY basis exists. Furthermore, when given a candidate basis B, if Algorithm 2 fails, then clearly no set of permutations Π exists such that B^Π is an NLY basis. In one case this is because there are three edges connected to a vertex by a different label, and thus the label 2 cannot be removed by any permutation. In the other it is because there is no solution to the 2-XOR-SAT problem, which rules out all potential permutations for those vertices connected to exactly two labels, while in the case of vertices connected to exactly one label, there are other possible permutations, but they would have the same action, and are thus also ruled out.

The only nontrivial fact left to prove is that if, given a candidate basis B, Algorithm 2 fails, then no NLY basis exists. Naively once could imagine that, given some alternative candidate basis, Algorithm 2 might succeed. Here we prove that this cannot happen, using proof by contradiction.

Algorithm 2: Algorithm for finding permutations Π such that B^{Π} is an NLY basis

Input : Graph $G = (V, E)$, rank-1 matrix edge weights $\{\beta_{uv}\}$, candidate basis $B = \{b_u\}$.

Output: A set of permutations $\Pi = \{\pi_u\}$ such that B^{Π} is an NLY basis, if one exists. Otherwise False.

for $v \in V$ **do**
  /* Label all incident edges according to which basis vector in b_v is not in the null space of β_{uv}. These always exist by Proposition 6.4. */
  $L(v) = \{\}$
  for $u \in V$ *such that* $e = (u, v) \in E$ **do**
    for $i \in \{1, 2, 3\}$ **do**
      $e_i^v = b_v[i]$
      if $\beta_{uv} e_i^v \neq 0$ **then**
        $L(v, e) = i$
        $L(v) = L(v) \cup \{i\}$
  /* If a vertex is incident on more than two different labels, then return False. */
  if $|L(v)| = 3$ **then**
    return False
  /* If the algorithm has not terminated, then for every vertex v there exist at most two labels such that every edge incident on v connects to v with one of those two labels. */
  /* Define permutations so that all incident edge labels are mapped to either 1 or 3. */
  Choose permutation π_v^0 such that $\pi_v^0(L(v)[1]) = 1$, and if $|L(v)| > 1$, then $\pi_v^0(L(v)[2]) = 3$.
  Choose permutation π_v^1 such that $\pi_v^1(L(v)[1]) = 3$, and if $|L(v)| > 1$, then $\pi_v^1(L(v)[2]) = 1$.
/* For each edge define a 2-XOR-SAT clause. */
for $e = (u, v) \in E$ **do**
  if $L(v, e) = L(u, e)$ **then**
    $C_e(x_u, x_v) = x_u \oplus x_v$
  else
    $C_e(x_u, x_v) = \bar{x}_u \oplus x_v$
/* The solution to the associated 2-XOR-SAT problem specifies which permutations to apply at each vertex so that $\pi_u^{x_u}(i) = \pi_v^{x_v}(j)$. By virtue of π never mapping any label to the value 2, and ensuring the uniform bilabeling of each edge, such an assignment will specify an NLY basis. */
success= 2-XOR-SAT(ref $\{x_v \mid \forall v \in V\}$, $\{C_e \mid \forall e \in E\}$)
if *success* **then**
  return $\Pi = \{\pi_v^{x_v} \mid \forall v \in V\}$
else
  return False

Assume that, given a candidate basis B, Algorithm 2 fails and there does not exist a permutation Π such that the basis B^{Π} is an NLY basis. Suppose, however, that there exists an NLY basis $\bar{B}$. If for some edge (u, v) adjacent to u the basis vector $e_i^u \in b_u$ satisfies $\beta_{uv}^{\mathsf{T}} e_i^u \neq 0$, then there must exist a unique vector $\bar{e}_j^u \in \bar{b}_u$ such that $\beta_{uv}^{\mathsf{T}} \bar{e}_j^u \neq 0$. Furthermore $e_i^u = \pm \bar{e}_j^u$, since β_{uv} is rank-1. Therefore for every index $i \in \{1, 2, 3\}$ there exists an index $j \in \{1, 2, 3\}$ such that for every edge $e = (u, v)$ adjacent to u, if e_i^u satisfies $\beta_{uv}^{\mathsf{T}} e_i^u \neq 0$, then $\bar{e}_j^u$ satisfies $\beta_{uv}^{\mathsf{T}} \bar{e}_j^u \neq 0$. Let π_u be the permutation with the mapping: $\pi_u(i) = j$, and $\Pi = \{\pi_u\}$. Then the bilabeled graph associated with B^{Π} must be identical to the bilabeled graph associated with $\bar{B}$, and therefore B^{Π} must be an NLY basis, which is a contradiction. ∎

6.4. Graphs with both rank > 1 and rank-1 edges. We will now show how the intuition and arguments given in subsection 6.3 translate into the case where the matrix weights may have any rank. First we outline the structure of the argument. Just as in subsection 6.3, we will first search for a candidate basis B for the graph, and then search for an appropriate set of permutations Π to apply to the basis vectors.

DEFINITION 6.8 (candidate basis of a graph). *A* candidate basis $B = \{b_u\}$ *is an assignment of basis vectors* $b_u = (e_1^u, e_2^u, e_3^u)$*, to each vertex* u*, satisfying the following two conditions:*

1. *For every rank-*1 *edge* (u, v) *adjacent to the vertex* u*, the basis vectors* b_u *are eigenvectors of* $\beta_{uv}\beta_{uv}^{\mathsf{T}}$.
2. *For every rank* > 1 *edge* (u, v)*, the basis vectors of* b_u *and* b_v *are left and right singular vectors of* β_{uv}*, respectively, and satisfy the following:* $\exists \sigma \in \mathbb{R}$ *such that* $\beta_{uv} e_i^v = \pm \sigma e_i^u$ *and* $\beta_{uv}^{\mathsf{T}} e_i^u = \pm \sigma e_i^v$.

The candidate basis has the same requirements on rank-1 edges as in the previous section; however, it satisfies more stringent requirements on rank > 1 edges, namely, that the transformed matrix weights $O_u^{\mathsf{T}} \beta_{uv} O_v$ are diagonal under the prescribed orthogonal rotations $\{O_u\}$. The most significant difference between the algorithms presented in this section is the procedure for finding a candidate basis (Algorithm 3). However, once a candidate basis has been found, the procedure for finding an appropriate set of permutations (Algorithm 4) will have the same essential form as Algorithm 2 with one difference: Instead of individual vertices being the sites to which permutations are assigned, we will instead assign permutations to subgraphs whose vertices are connected by rank > 1 paths (Definition 6.11), so that each vertex in such a subgraph is permuted uniformly. This is illustrated in Figure 4, in contrast to Figure 3. It will be straightforward to see that if Algorithms 3 and 4 succeed, then they will have produced an NLY basis. The only significant subtle point that remains, and will be argued in Theorem 6.14, is that if Algorithm 4 is given a candidate basis and fails to find a set of permutations which produces an NLY basis, then no NLY basis exists, and in particular no other candidate bases need to be considered.

Before proceeding with the description of the algorithm for finding a candidate basis, we must establish some facts about rank > 1 edges, and the structure they impose on the problem.

LEMMA 6.9. *Given a rank* > 1 *edge* (u, v)*, and bases* b_u*,* b_v *which are eigenvectors of* $\beta_{uv}\beta_{uv}^{\mathsf{T}}$ *and* $\beta_{uv}^{\mathsf{T}}\beta_{uv}$*, respectively, the vectors* $e_i^u \in b_u$ *and* $e_i^v \in b_v$ *satisfy* $\beta_{uv} e_i^v = \pm \sigma_i e_i^u$ *and* $\beta_{uv}^{\mathsf{T}} e_i^u = \pm \sigma_i e_i^v$*,* $\sigma_i \in \mathbb{R}$*, if and only if, for every singular value decomposition* $\beta_{uv} = O_u^e \Sigma_{uv}^{\mathrm{SVD}} (O_v^e)^{\mathsf{T}}$*, the operator defined as*

$$O_{v \leftarrow u} = O_{u \leftarrow v}^{\mathsf{T}} := O_v^e (O_u^e)^{\mathsf{T}} \tag{6.9}$$

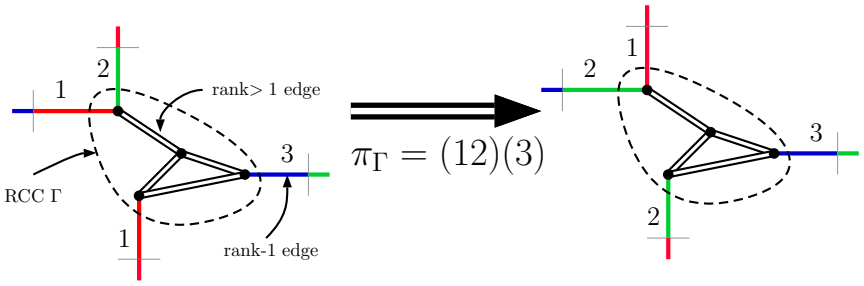

FIG. 4. *Action of a permutation* (12)(3) *on a rank* > 1 *connected component (RCC) in black.*

satisfies $O_{v\leftarrow u}e_i^u = \pm e_i^v$, $O_{u\leftarrow v}e_i^v = \pm e_i^u$ $\forall i$. *In other words,*

(6.10) $O_{v\leftarrow u}b_u = b_v$ *modulo signs, and equivalently* $O_{u\leftarrow v}b_v = b_u$ *modulo signs.*

Proof. First we prove the *only if* condition. Given that $\beta_{uv}^{\mathsf{T}}e_i^u = \pm\sigma_i e_i^v$ and $\beta_{uv}\beta_{uv}^{\mathsf{T}}e_i^u = \sigma_i^2 e_i^u$ we have

$$O_v^e\Sigma_{uv}^{\mathrm{SVD}}(O_u^e)^{\mathsf{T}}e_i^u = \pm\sigma_i e_i^v\ ,\ \ (\Sigma_{uv}^{\mathrm{SVD}})^2(O_u^e)^{\mathsf{T}}e_i^u = \sigma_i^2(O_u^e)^{\mathsf{T}}e_i^u.$$

If a real matrix A is nonnegative and diagonal, then any eigenvectors of A^2 with eigenvalues λ are also the eigenvectors of A with eigenvalues $|\sqrt{\lambda}|$. Since $\Sigma_{uv}^{\mathrm{SVD}}$ is nonnegative and diagonal, we see that $(O_u^e)^{\mathsf{T}}e_i^u$ is an eigenvector of $\Sigma_{uv}^{\mathrm{SVD}}$ with eigenvalue $|\sigma_i|$. It follows that

$$O_v^e\Sigma_{uv}^{\mathrm{SVD}}(O_u^e)^{\mathsf{T}}e_i^u = \pm\sigma_i e_i^v \Rightarrow |\sigma_i|O_v^e(O_u^e)^{\mathsf{T}}e_i^u = \pm\sigma_i e_i^v.$$

If $\sigma_i \neq 0$, then $O_v^e(O_u^e)^{\mathsf{T}}e_i^u = \pm e_i^v$. Suppose $\sigma_i = 0$; then since $\mathrm{Rank}(\beta_{uv}) > 1$ there is a single i for which this holds. Since $\forall j \neq i$, $O_v^e(O_u^e)^{\mathsf{T}}e_j^u = \pm e_j^v$, it follows that e_i^v must lie in the one-dimensional subspace spanned by $O_v^e(O_u^e)^{\mathsf{T}}e_i^u$, and so $O_v^e(O_u^e)^{\mathsf{T}}e_i^u = \pm e_i^v$.

Now we prove the *if* condition. Given that $\beta_{uv}\beta_{uv}^{\mathsf{T}}e_i^u = \sigma_i^2 e_i^u$ and $O_v^e(O_u^e)^{\mathsf{T}}e_i^u = \pm e_i^v$,

$$\begin{aligned}\beta_{uv}\beta_{uv}^{\mathsf{T}}e_i^u &= \sigma_i^2 e_i^u,\\ O_u^e\Sigma_{uv}^{\mathrm{SVD}}(O_v^e)^{\mathsf{T}}O_v^e\Sigma_{uv}^{\mathrm{SVD}}(O_u^e)^{\mathsf{T}}e_i^u &= \sigma_i^2 e_i^u,\\ (\Sigma_{uv}^{\mathrm{SVD}})^2(O_u^e)^{\mathsf{T}}e_i^u &= \sigma_i^2(O_u^e)^{\mathsf{T}}e_i^u.\end{aligned}$$

Since $\Sigma_{uv}^{\mathrm{SVD}}$ is a positive diagonal matrix we have

$$\begin{aligned}(\Sigma_{uv}^{\mathrm{SVD}})(O_u^e)^{\mathsf{T}}e_i^u &= |\sigma_i|(O_u^e)^{\mathsf{T}}e_i^u,\\ O_v^e(\Sigma_{uv}^{\mathrm{SVD}})(O_u^e)^{\mathsf{T}}e_i^u &= |\sigma_i|O_v^e(O_u^e)^{\mathsf{T}}e_i^u,\\ \beta_{uv}^{\mathsf{T}}e_i^u &= \pm\sigma_i e_i^v.\end{aligned}$$

By a symmetric argument $\beta_{uv}e_i^v = \pm\sigma_i e_i^u$. ∎

It is important to note that the construction of $O_{v\leftarrow u}$ in (6.9) is not unique. One could find a different singular value decomposition and construct a different operator $O'_{v\leftarrow u}$. However, as proven above, for any such operator its action on a singular vector e_i^u of β_{uv} is identical, up to a difference in the sign, which has no bearing on

the problem. In light of this, for the remainder of the text we will treat the operator $O_{v\leftarrow u}$ as a well-defined orthogonal operator, with the implicit assumption being that any such operator suffices.

The above lemma has two important consequences.

COROLLARY 6.10.

1. *Condition* 2 *in Definition* 6.8 *is equivalent to the condition that for every rank* > 1 *edge* (u, v) *and any operator* $O_{v\leftarrow u}$,
$$O_{v\leftarrow u} b_u = b_v \text{ modulo signs.}$$
2. *If* $B = \{b_u\}$ *is a candidate basis, then given a path* $p = (u, x, \ldots, y, w, v)$ *of rank* > 1 *edges going from vertex* u *to* v, *for any orthogonal rotation defined by* $O_p = O_{v\leftarrow w} O_{w\leftarrow y} \ldots O_{x\leftarrow u}$, *which we call a* rank > 1 path operator, *it must be the case that* $O_p b_u = b_v$ *modulo signs.*

Proof. The first statement follows trivially by the definition of the candidate basis. The second statement follows by induction from the first. ∎

We see that if for some vertex u we choose a basis b_u which happens to belong to a yet unknown candidate basis B, then this fixes, via the rank > 1 path operators, all of the bases $b_v \in B$, modulo signs, for all vertices v connected to u by rank > 1 paths. This motivates the following definition.

DEFINITION 6.11 (rank > 1 connected component (RCC)). *Remove all rank-*1 *edges from the graph* G. *What remains is a family of distinct connected components which are composed entirely of rank* > 1 *edges. Define the* rank > 1 connected component (RCC) *as the subgraph* Γ *associated with such a connected component.*

Note that in the case where some vertex v is connected to only rank-1 edges, v on its own still constitutes an RCC. Therefore by construction every vertex is in exactly one RCC. Note also that any two vertices connected by a path of rank > 1 edges belong to the same RCC.

DEFINITION 6.12 (candidate basis on an RCC). *Given an RCC* Γ, *a* candidate basis of an RCC B_Γ *on the vertices of* Γ *is the assignment of a basis* b_u *to each vertex* $u \in \Gamma$ *which satisfies all the conditions of a candidate basis for all of the rank* > 1 *edges in* Γ, *as well as for the rank-*1 *edges (which are not in* Γ*) which are adjacent to the vertices in* Γ.

Clearly, if we combine all the candidate bases for the RCCs Γ, we obtain a candidate basis for the vertices of the whole graph G. So the task of finding a candidate basis B for the whole graph breaks down into finding a candidate basis B_Γ for each RCC Γ, so that $B = \bigcup_\Gamma B_\Gamma$. Furthermore, if we can correctly choose a basis b_u at one vertex u in Γ, then, due to Corollary 6.10, we will have successfully specified all of B_Γ. The primary challenge is making the right choice of b_u.

LEMMA 6.13. *Given a matrix-weighted graph, Algorithm* 3 *efficiently finds a candidate basis* B *or otherwise shows that no such candidate basis exists. The algorithm takes* $O(N^3)$ *steps, where* N *is the number of vertices.*

Proof. Let us first prove that if the algorithm returns B, then B is a candidate basis. To show that B is a candidate basis, we need only show that $\forall \Gamma$, B_Γ is a candidate basis.

The first fact to note is that for every vertex $v \in \Gamma$, and for every edge (v, w) adjacent to v, the basis vectors b_v are eigenvectors of $\beta_{vw}\beta_{vw}^{\mathsf{T}}$, and so the first condition

Algorithm 3: Algorithm for finding a candidate basis

Input : Graph $G = (V, E)$, matrix edge weights $\{\beta_{uv}\}$, RCCs $\{\Gamma\}$.
Output: A candidate basis $B = \{b_u\}$, if one exists. Otherwise False, indicating no candidate basis exists.

for $\Gamma \in$ **RCCs do**
  /* Construct the $O_{v\leftarrow u}$ operators for all the edges in Γ */
  for $e = (u, v) \in \Gamma$ **do**
    $(O_u^e, \Sigma_{uv}^{\text{SVD}}, O_v^e) = \text{SVD}(\beta_{uv})$ s.t. $O_u^e \Sigma_{uv}^{\text{SVD}} (O_v^e)^T = \beta_{uv}$
    $O_{v\leftarrow u} = O_v^e (O_u^e)^{\mathsf{T}}$
    $O_{u\leftarrow v} = O_{v\leftarrow u}^{\mathsf{T}}$
  /* At each vertex take intersection of eigenspaces associated with immediately neighboring edges, as in Algorithm 1. */
  for $v \in \Gamma$ **do**
    $\mathbb{S}_v[0] = \{\mathbb{R}^3\}$
    for $u \in V$ *such that* $e = (u, v) \in E$ **do**
      $\mathbb{S}_v^e =$ the set of orthogonal maximal eigenspaces of $\beta_{uv}^{\mathsf{T}} \beta_{uv}$
      $\mathbb{S}_v[0] = \mathbb{S}_v[0] \cap \mathbb{S}_v^e$
  /* For each vertex, iteratively take the intersection of the subspaces at that vertex, with the appropriately rotated subspaces of the neighboring vertices. */
  $i = 0$
  while *True* **do**
    for $v \in \Gamma$ **do**
      $\mathbb{S}_{\text{neighbors}} = \{\mathbb{R}^3\}$
      for $u \in \Gamma$ *such that* $(u, v) \in \Gamma$ **do**
        $\mathbb{S}_{\text{neighbors}} = \mathbb{S}_{\text{neighbors}} \cap (O_{v\leftarrow u} \mathbb{S}_u[i])$
      $\mathbb{S}_v[i+1] = \mathbb{S}_v[i] \cap \mathbb{S}_{\text{neighbors}}$
    /* Conclude the iterative process when it reaches a fixed point. Note that since every subspace in $\mathbb{S}_v[i+1]$ is contained in a subspace of $\mathbb{S}_v[i]$, this process must reach a fixed point. */
    if $\mathbb{S}_v[i+1] = \mathbb{S}_v[i] \ \ \forall v \in \Gamma$ **then**
      $\mathbb{S}_v[f] := \mathbb{S}_v[i] \ \ \forall v \in \Gamma$
      break
    i++
  /* Choose a spanning tree and construct the intersection of the eigenspaces of the rank > 1 path operators associated with the fundamental cycles. Take the intersection of this with the set of subspaces at the root vertex. */
  $T =$ spanning tree of Γ with root vertex r
  $\mathbb{S}_{\text{loops}} = \{\mathbb{R}^3\}$

[*Continued below*]

necessary for B_Γ to be a candidate basis is satisfied. To see that this is true, it suffices to show that $b_v \in \mathbb{S}_v[f]$, since $\mathbb{S}_v[f] \subseteq \mathbb{S}_v[0]$, and $\mathbb{S}_v[0]$ by construction only contains eigenvectors of all neighboring edges, including rank-1 edges. For all $w \in \Gamma$, since

Algorithm 3 continued

```
24  for edge e ∈ Γ such that e ∉ T do
25      Get C_e, the fundamental cycle associated with e.
26      Let p_e = (r, u, v, ..., w, r) be the ordered vertex sequence of C_e.
27      Construct O_{p_e} = O_{r←u} O_{u←v} ... O_{w←r}, the associated rank > 1 path
          operator.
28      Find {λ_i} and 𝕊_{p_e} = {S_i^{p_e}}, the eigenvalues and maximal eigenspaces
          of the orthogonal matrix O_{p_e}. (Note that every orthogonal matrix is
          diagonalizable.)
29      if ∃i such that λ_i ∉ ℝ then
30          return False
31      𝕊_loops = 𝕊_loops ∩ 𝕊_{p_e}
32  𝕊*_r = 𝕊_r[f] ∩ 𝕊_loops
33  if span(𝕊*_r) ≠ ℝ^3 then
34      return False
    /* Note that the intersection of sets of orthogonal subspaces
       is also a set of orthogonal subspaces.  Thus if
       span(𝕊*_r) = ℝ^3, then one can always choose an orthogonal
       basis from it.                                              */
35  Choose orthonormal basis b_r = (e_1^r, e_2^r, e_3^r) ⊆ 𝕊*_r.
    /* Propagate the choice of basis at the root vertex out to
       the rest of the vertices in the tree.                       */
36  def propagate(u, b_u):
37      for vertex v ∈ T that are children of u do
38          b_v = O_{v←u} b_u
39          propagate(v, b_v)
40  propagate (r, b_r)
41  B_Γ = {b_u : ∀u ∈ Γ}
42 return B = ⋃_Γ B_Γ
```

$\mathbb{S}_w[f]$ is a fixed point of the equation on line 17 of Algorithm 3, we have

$$\mathbb{S}_w[f] = \mathbb{S}_w[f] \cap \left(\bigcap_x O_{w \leftarrow x} \mathbb{S}_x[f] \right), \tag{6.11}$$

where x runs over rank > 1 edges adjacent to w. Thus for all rank > 1 edges (w, x) in Γ, $\mathbb{S}_w[f] \subseteq O_{w \leftarrow x} \mathbb{S}_x[f]$. Consequently, given a vertex w in the spanning tree T, and a child vertex x, if $b_w \subseteq \mathbb{S}_w[f]$, then since $b_x = O_{x \leftarrow w} b_w \Rightarrow O_{w \leftarrow x} b_x = b_w$ it follows that $b_x \subseteq \mathbb{S}_x[f]$. Since $b_r \subseteq \mathbb{S}_r[f]$, and r is the root node of T, it follows by induction that $\forall v \in \Gamma$, $b_v \subseteq \mathbb{S}_v[f]$.

The second fact to note is that for every rank > 1 edge (w, v) in Γ, $b_w = O_{w \leftarrow v} b_v$, modulo signs. This is clearly true for every rank > 1 edge in T by construction, as specified in line 38 of Algorithm 3. All that remains are those rank > 1 edges not in T. Consider an edge $e = (v, w)$ not in T. There is a fundamental cycle C_e, with a path p_e which goes from the root vertex r, up to v, entirely along paths in the spanning tree, then from v to w, and then from w back to r. Thus the associated

rank > 1 path operator is

$$O_{p_e} = O_{r\leftarrow x} \dots O_{y\leftarrow w} O_{w\leftarrow v} O_{v\leftarrow z} \dots O_{q\leftarrow r}.$$

Furthermore, the bases b_v and b_w are, by construction,

$$b_v = O_{v\leftarrow z} \dots O_{q\leftarrow r} b_r,$$

$$b_w = O_{w\leftarrow y} \dots O_{x\leftarrow r} b_r \rightarrow b_w = O^{\mathsf{T}}_{y\leftarrow w} \dots O^{\mathsf{T}}_{r\leftarrow x} b_r.$$

By construction, every element in b_r must be an eigenvector of O_{p_e} with real eigenvalues (equal to $+1$ or -1 since O_{p_e} is an orthogonal matrix) (see line 35 of Algorithm 3). Thus $b_r = O_{p_e} b_r$ modulo signs. Therefore

$$\begin{aligned} b_r &= O_{r\leftarrow x} \dots O_{y\leftarrow w} O_{w\leftarrow v} O_{v\leftarrow z} \dots O_{q\leftarrow r} b_r \text{ modulo signs,} \\ O^{\mathsf{T}}_{y\leftarrow w} \dots O^{\mathsf{T}}_{r\leftarrow x} b_r &= O_{w\leftarrow v} O_{v\leftarrow z} \dots O_{q\leftarrow r} b_r \text{ modulo signs,} \\ b_w &= O_{w\leftarrow v} b_v \quad \text{modulo signs.} \end{aligned}$$

Thus for every rank > 1 edge (w, v) in Γ, $b_w = O_{w\leftarrow v} b_v$, modulo signs. Combining this fact with claim 1 of Corollary 6.10, it is clear that the second condition necessary for B_Γ to be a candidate basis is satisfied. Therefore B_Γ is a candidate basis.

Now we will prove that if Algorithm 3 returns False, then no candidate basis exists. First we note that obviously if for any Γ there does not exist a candidate basis B_Γ, then no candidate basis exists for the whole graph.

There are two places where the algorithm returns False: once on line 30, and once on line 34. This happens on line 30 if some O_{p_e} has any nonreal eigenvalues. Note that by claim 2 in Corollary 6.10, if there existed a candidate basis $B = \{b_u\}$, then $O_{p_e} b_r = b_r$, modulo signs, since O_{p_e} is a rank > 1 path operator. In other words, the eigenvalues of O_{p_e} should be either $+1$ or -1.[3]

The algorithm indicates on line 34 that no candidate basis exists if, for a given RCC Γ with root vertex r, span $(\mathbb{S}^*_r) \neq \mathbb{R}^3$. This happens if and only if there does not exist a set of three orthogonal vectors such that each of them belongs to a subspace in $\mathbb{S}_r[f]$ as well as a subspace in every $\mathbb{S}_{p_e}$. We prove by contradiction that in this case B_Γ must not exist.

Suppose there does not exist a set of three orthogonal vectors such that each of them belongs to a subspace in $\mathbb{S}_r[f]$ as well as a subspace in every $\mathbb{S}_{p_e}$. Suppose a candidate basis B_Γ does exist; then by the argument made for the case of line 30 in the preceding paragraph, the basis vectors b_r must be eigenvectors of every O_{p_e}, and thus each vector in b_r belongs to a subspace in $\mathbb{S}_{p_e}$ for every p_e. Therefore it must be that $b_r \not\subseteq \mathbb{S}_r[f]$. However, this is contradicted by the following argument.

First note that $\forall b_v \in B_\Gamma$ and for every adjacent edge (v, x), the basis vectors b_v must be eigenvectors of $\beta_{vx}\beta^{\mathsf{T}}_{vx}$, and thus $b_v \subseteq \mathbb{S}_v[0]$. Second note that if $\forall b_v \in B_\Gamma$, $b_v \subseteq \mathbb{S}_v[i]$, then $\forall b_v \in B_\Gamma$, $b_v \subseteq \mathbb{S}_v[i+1]$. This follows from the fact that for all vertices $v \in \Gamma$, and for all rank > 1 edges (v, x) adjacent to v, $b_v = O_{v\leftarrow x} b_x$, modulo signs, by Corollary 6.10, and thus $b_v \subseteq O_{v\leftarrow x} \mathbb{S}_x[i]$. Since $\mathbb{S}_v[i+1] = \mathbb{S}_v[i] \cap (\bigcap_x O_{v\leftarrow x} \mathbb{S}_x[i])$ it follows that $b_v \subseteq \mathbb{S}_v[i+1]$. Thus, by induction, $\forall b_u \in B_\Gamma$, $b_u \subseteq \mathbb{S}_u[f]$, which is a contradiction.

[3]For an example of where such a loop operation becomes important, see Appendix C.

Finally we prove that the algorithm runs in $O(N^3)$ steps, where N is the number of vertices in the graph. The most costly part of the algorithm is the while loop on line 12. Let n_Γ be the number of vertices in the RCC Γ. Constructing all $\mathbb{S}_u[0]$ runs in worst case $O(n_\Gamma N)$. Each iterative step runs in worst case $O(n_\Gamma N)$. At each iterative step the subspaces of $\mathbb{S}_u[i+1]$ must be contained in the subspaces in $S_u[i]$. Therefore if we have not reached a fixed point, then at each iterative step there is at least one $\mathbb{S}_u[i+1]$ for which the dimensions of the subspaces have decreased when compared to $\mathbb{S}_u[i]$. If $\mathbb{S}_u[i]$ spans $\mathbb{R}^3$, then the dimensions of its mutually orthogonal subspaces must be either (3), $(2,1)$, or $(1,1,1)$. Thus for every vertex u, the iterative process can only decrease the dimensionality of the subspaces in $\mathbb{S}_u[i]$ at most three times before $\mathbb{S}_u[i]$ no longer spans $\mathbb{R}^3$. So the maximum number of iterations is $3n_\Gamma$.

Therefore the naive worst case runtime of this step is $O(n_\Gamma^2 N)$. However, we expect that a more careful analysis would find the runtime to be closer to $O(n_\Gamma N)$, since the runtime of each iterative step is proportional to the connectivity of the graph, while the number of iterative steps required should be inversely proportional to the connectivity.

On line 24 the algorithm iterates over edges in Γ, not in T, the number of which is upper bounded by $O(n_\Gamma^2)$. All other steps in the algorithm iterate over vertices in Γ, or edges adjacent to those vertices, and so have runtime at most $O(n_\Gamma N)$.

Since the whole algorithm iterates over all RCCs, it follows that the runtime is $O\left(\sum_\Gamma n_\Gamma^2 N\right)$, which, by the triangle inequality, is upper bounded by $O(N^3)$. ∎

THEOREM 6.14. *Given a matrix-weighted graph, one can efficiently find an NLY basis, or else show that no such basis exists.*

Proof. The procedure for finding an NLY basis is to first find a candidate basis B using Algorithm 3, and then find a set of permutations Π such that B^Π is an NLY basis using Algorithm 4.

If Algorithm 4 is successful, then B^Π is an NLY basis by the following reasoning. For every rank > 1 edge an identical permutation is applied to its adjacent vertices, and so the diagonality of the matrix weights is preserved, while for the rank-1 edges, the matrix weights are diagonal, and by construction every matrix weight is zero in its second diagonal entry, as per arguments made in subsection 6.3.

Algorithm 4 is efficient, since the number of variables in the 2-XOR-SAT problem is the number of RCCs, and in the worst case this is the number of vertices N, so it runs in time $O(N^2)$. Combining this with Lemma 6.13, the worst case runtime of the whole algorithm is $O(N^3)$.

Finally, if either of these algorithms fails, then by the following two arguments we claim that no NLY basis exists. First, if Algorithm 3 fails, then by Lemma 6.13 no candidate basis exists, and since an NLY basis must satisfy the conditions for being a candidate basis, no NLY basis exists. Second, we must establish the nontrivial fact that if Algorithm 4 fails, then no NLY basis exists. In other words, we need to rule out the possibility that Algorithm 4 might have succeeded had we supplied it with an alternative candidate basis. The rest of our exposition is devoted to proving this fact.

First note that, given a candidate basis B, if Algorithm 4 fails, then no set of permutations exists such that B^Π is an NLY basis. This follows from the fact that if a permutation were to exist, it must be uniform on every RCC in order to preserve the diagonality of the rank > 1 edges. Given this, the argument reduces to the same one made in Theorem 6.7, where we treat RCCs as sites, since every rank-1 edge is adjacent to RCCs.

Algorithm 4: Algorithm for finding permutations Π such that B^{Π} is an NLY basis

Input : Graph $G = (V, E)$, rank > 1, and rank-1 matrix edge weights $\{\beta_{uv}\}$, RCCs $\{\Gamma\}$, and candidate basis $B = \{b_u\}$.

Output: A set of permutations $\Pi = \{\pi_u\}$ such that B^{Π} is an NLY basis, if one exists. Otherwise False, indicating none exists.

for $\Gamma \in$ *RCCs* **do**
  /* Label all rank-1 edges incident to vertices v in Γ according to which basis vector b_v is not in the null space of β_{uv}. These always exist by Proposition 6.4. */
  $L(\Gamma) = \{\}$
  for $e = (u, v) \in E$ *such that* $v \in \Gamma$, *and rank*$(\beta_{uv}) = 1$ **do**
    for $i \in \{1, 2, 3\}$ **do**
      $e_i^u = b_u[i]$
      if $\beta_{uv} e_i^v \neq 0$ **then**
        $L(v, e) = i$
        $L(\Gamma) = L(\Gamma) \cup \{i\}$
  /* If Γ is incident on more than two different labels, then return False. */
  if $|L(\Gamma)| = 3$ **then**
    return False
  /* Define permutations so that all incident edge labels are mapped to either 1 or 3. */
  Choose perm. π_Γ^0 such that $\pi_\Gamma^0(L(\Gamma)[1]) = 1$, and if $|L(\Gamma)| > 1$, then $\pi_\Gamma^0(L(\Gamma)[2]) = 3$.
  Choose perm. π_Γ^1 such that $\pi_\Gamma^1(L(\Gamma)[1]) = 3$, and if $|L(\Gamma)| > 1$, then $\pi_\Gamma^1(L(\Gamma)[2]) = 1$.
/* The task now becomes assigning a binary value x_Γ to each Γ so that for every rank-1 edge $e = (u, v)$, with labels $L(u, e) = i$ and $L(v, e) = j$, which has vertices in Γ_u and Γ_v, the binary assignment x_{Γ_u} to Γ_u and x_{Γ_v} to Γ_v satisfies $\pi_{\Gamma_u}^{x_{\Gamma_u}}(i) = \pi_{\Gamma_v}^{x_{\Gamma_v}}(j)$. */
/* For each rank-1 edge define a 2-XOR-SAT clause on Boolean variables associated with the RCCs on which the edge is incident. */
for $e = (u, v) \in E$ *such that* $\operatorname{rank}(\beta_{uv}) = 1$ **do**
  Find $\Gamma_u \in$ RCCs such that $u \in \Gamma_u$.
  Find $\Gamma_v \in$ RCCs such that $v \in \Gamma_v$.
  if $L(v, e) = L(u, e)$ **then**
    $C_e(x_{\Gamma_u}, x_{\Gamma_v}) = x_{\Gamma_u} \oplus x_{\Gamma_v}$
  else
    $C_e(x_{\Gamma_u}, x_{\Gamma_v}) = \bar{x}_{\Gamma_u} \oplus x_{\Gamma_v}$

[*Continued below*]

Algorithm 4 continued

```
/* The solution to the associated 2-XOR-SAT problem specifies,
   for each RCC Γ, which permutations to apply in a uniform
   fashion to all vertices in Γ.                              */
variables = {x_Γ | ∀Γ ∈ RCCs}
clauses = {C_e | ∀e = (u, v) ∈ E such that rank(β_uv) = 1}
success= 2-XOR-SAT(ref variables, clauses)
if ¬success then
    return False
for Γ ∈ RCCs do
    for vertex v ∈ Γ do
        π_v = π_Γ^{x_Γ}
return Π = {π_v | ∀v ∈ V}
```

Given that if the procedure in Algorithm 4 fails, then there does not exist a permutation Π such that the basis B^{Π} is an NLY basis, we use proof by contradiction to show that in this case no NLY basis exists.

Suppose there exists an NLY basis $\bar{B} = \{\bar{b}_u\}$. We now argue that for a fixed RCC Γ, for every index $i \in \{1, 2, 3\}$ there exists an index $j \in \{1, 2, 3\}$ such that for every vertex $u \in \Gamma$, if $e_i^u \in b_u$ corresponds to a left singular vector, with nonzero singular value, of a rank-1 edge adjacent to u, then $e_i^u = \pm\bar{e}_j^u \in \bar{b}_u$.

Given an index i, consider any two vertices $u, v \in \Gamma$ for which e_i^u, e_i^v correspond to singular vectors, with nonzero singular values, of some rank-1 edges adjacent to u and v. Consider that $\bar{b}_u$ must also contain a vector $\bar{e}_{j_u}^u$ at some particular index j_u, which is also a singular vector with nonzero singular value of the same rank-1 edge adjacent to u. Since that edge is rank-1, it follows that $e_i^u = \pm\bar{e}_{j_u}^u$. A similar argument can be made for v so that $e_i^v = \pm\bar{e}_{j_v}^v$ for some index j_v. There must exist a rank > 1 path p connecting u to v, and by Corollary 6.10, $O_p e_i^u = \pm e_i^v$. Similarly, $O_p \bar{e}_{j_u}^u = \pm\bar{e}_{j_u}^v$. Therefore $\pm\bar{e}_{j_u}^v = \bar{e}_{j_v}^v$ and since each vector in $\bar{b}_v$ is orthogonal we have $j_u = j_v = j$. Since this is true for any such pair of vertices u, v, there must exist an index j such that for every vertex $u \in \Gamma$, if e_i^u corresponds to a left singular vector, with nonzero singular value, of a rank-1 edge adjacent to u, then $e_i^u = \pm\bar{e}_j^u \in \bar{b}_u$.

We can now proceed by the same reasoning used in the proof of Theorem 6.7. Let π_Γ be the permutation with the mapping $\pi_\Gamma(i) = j$, and let $\pi_u = \pi_\Gamma \ \forall u \in \Gamma$, and define $\Pi = \{\pi_u\}$. Then the bilabeled graph associated with B^{Π} must be identical to the bilabeled graph associated with $\bar{B}$ (i.e., the action on the rank-1 edges is the same), and therefore B^{Π} must be an NLY basis, which is a contradiction. ∎

7. Discussion. It is clear from the work presented here that in the case of two-local qubit Hamiltonians, the hardness of curing the sign problem by local basis transformations is determined by the presence or absence of one-local terms in the Hamiltonian.

The question of whether the general LocalSignCure is a problem in NP for general two-local n-qubit Hamiltonians is not clear, as the set of local unitary transformations is a continuous parameter space, and a prover would need to specify a sign-curing solution with a polynomial number of bits, and such exact sign-curing transformations may not exist. A natural relaxation would be to demand that the transformation be

approximately sign-curing, a direction of research that is explored in [15], so that the problem would be contained in MA.

We do not know the complexity of determining the ground state energy for the family of Hamiltonians presented in section 5. It may be that finding the ground state energy is easy, thus obviating any interest in a curing transformation. It would be interesting to show that a family of Hamiltonians exists for which deciding sign-curing and determining ground state energy are both hard. It would be very surprising indeed if the hardness of deciding a curing transformation only appears when the ground state energy is efficiently computable.

A natural extension of sign-curing transformations beyond single-qubit unitary transformations is transformations which first embed each qubit into a d-dimensional system and then allow for local basis changes in this d-dimensional system. The power of such "lifting" basis changes is completely unexplored, even in the two-qubit case. Another class of sign-curing transformations are Clifford circuits which map a Hamiltonian composed of poly(n) k-local Pauli's onto a sum of poly(n) nonlocal Pauli's. The power of these transformations is also largely unexplored, but some first results are reported in [22]. Recently, it was demonstrated [19] that even when there is an essential sign problem in the Hamiltonian, there are ways to group terms in the expansion of the Gibbs state to avoid the sign problem. It would be interesting to better understand how these techniques relate to stoquastic Hamiltonians.

Another strand of interesting future research concerns the distinction between termwise and globally stoquastic Hamiltonians. Examples can be constructed of 3-local globally stoquastic but not termwise-stoquastic Hamiltonians, and the complexity of deciding global stoquasticity can be analyzed.

Appendix A. A simple example of nonstoquastic two-local Hamiltonian. Here we present a two-qubit Hamiltonian that cannot be transformed into a symmetric Z-matrix by any single-qubit unitary transformations. Consider the Hamiltonian

$$H = -ZZ - 2XX + 3YY + IX + IZ + ZI + XI.$$

The β-matrix of this Hamiltonian is of the form

$$\beta = \begin{pmatrix} -2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

First note that this Hamiltonian is not stoquastic in this basis because $a_{XX} > -|a_{YY}|$. The orthogonal rotations on the β-matrix must be confined to the XZ plane in order to avoid complex terms like IY or XY. Any pair of such orthogonal transformations (given by angles θ_1 and θ_2) will keep the β-matrix in a block-diagonal form, and the new a'_{XX} entry will be

$$a'_{XX} = -2\cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2) > -3.$$

Therefore $a'_{XX} > -|a_{YY}|$ for all values of θ_1 and θ_2, and so H cannot be transformed into a symmetric Z-matrix by single-qubit unitary transformations.

Appendix B. Curing the sign problem for strictly two-local Hamiltonians by single-qubit Clifford transformations is easy. Suppose that instead of single-qubit unitaries, one is interested in curing a strictly two-local Hamiltonian

by single-qubit Clifford transformations. Using arguments similar to those outlined in section 6, we now show that such a problem is easy.

First, we know that the transformations that are employed in the XYZ-algorithm are single-qubit Clifford transformations. Furthermore, single-qubit Clifford transformations correspond to signed permutations on the matrix-weighted graph. Therefore, by the same logic as for the one-local unitary case, it suffices to find an algorithm which answers Problem 1, where instead of searching for a set of orthogonal rotations $\{O_u\}$, one instead seeks a set of signed permutations $\{\Pi_u\}$.

Now, instead of being able to consider any basis $B = \{b_u\}$, we only consider bases which are related to the standard basis by signed permutations:

$$b_u = \{\Pi_u e_1, \Pi_u e_2, \Pi_u e_3\}.$$

Again, the signs are irrelevant, and therefore we are looking for bases which are related to the standard basis by a permutation.

We say a matrix is quasi-monomial if for each row and column of that matrix there is at most one nonzero entry.[4] A matrix β_{uv} for an edge $e = (u, v)$ which is quasi-monomial admits a singular value decomposition of the form

$$\beta_{uv} = \Pi_u \Sigma_{uv}^{\mathrm{SVD}} (\Pi_v)^{\mathsf{T}},$$

where Π_u and Π_v are signed permutations. This can be seen by noting that by an appropriate permutation of columns and rows, the nonzero entries of β_{uv} can be made positive and put on the diagonal in descending order, which corresponds to the singular value decomposition of β. It follows by definition that any $O_{v \leftarrow u}$, as defined in (6.9), is also a signed permutation. It is not difficult to see that if any edge in the matrix-weighted graph of our Hamiltonian has a weight β_{uv} which is not quasi-monomial, then there can be no set of signed permutations which simultaneously diagonalize the weights of the graph. This is because a diagonalized matrix is quasi-monomial, and it is impossible to transform a non-quasi-monomial matrix into a quasi-monomial matrix by permuting the rows and columns.

We now describe the algorithm for answering Problem 1 in the case where we are interested in signed permutations instead of orthogonal rotations. First check that every matrix weight β_{uv} is a quasi-monomial matrix, as this is a necessary condition by the arguments above. If any of these matrices are not, then we return False.

We then identify a candidate basis B_Γ for each RCC such that the bases $b_u \in B_\Gamma$ are permutations of the standard basis. For each RCC construct the set of subspaces $\mathbb{S}_u^*$ as described in Algorithm 3. Then check if the standard basis belongs to $\mathbb{S}_u^*$. By the same arguments made in Lemma 6.13 it is clear that if the standard basis is not in $\mathbb{S}_u^*$, then no permutations of the standard basis are in $\mathbb{S}_u^*$, and so no candidate basis exists for Γ which is a permutation of the standard basis, and so we must return False. If the standard basis is in $\mathbb{S}_u^*$, then we choose the standard basis for b_u, and construct a candidate basis B_Γ for Γ as per step 8 of Algorithm 3. Since every operator $O_{v \leftarrow u}$ is a signed permutation, it follows that all other bases $b_v \in B_\Gamma$ are permutations of the standard basis. Let $B = \bigcup_\Gamma B_\Gamma$.

Equipped now with a candidate basis for the graph, we can proceed with Algorithm 4. Noting that the only transformations being performed in this section are permutations on the candidate basis, we know that any NLY basis that is found will

[4] In a monomial matrix each row and column have exactly one nonzero entry, hence the "quasi-" prefix.

be a permutation of the standard basis. As such, whichever answer it gives will be the answer to our problem. ∎

We remark that even when the graph is weighted by only quasi-monomial matrices, it is generally not sufficient to consider only single-qubit Clifford transformations as curing transformations. This is proved in Appendix C.

Appendix C. Single-qubit Clifford transformations do not suffice to cure the sign problem for a quasi-monomial matrix-weighted graph. In this appendix we show that when the β-matrices associated with a graph are quasi-monomial, as introduced in Appendix B, then, even if there does not exist a set of signed permutations $\{\Pi_u\}$ such that $\Pi_u^\mathsf{T}\beta_{uv}\Pi_v$ is diagonal, there may still exist a set of orthogonal transformations $\{O_u\}$ such that $O_u^\mathsf{T}\beta_{uv}O_v$ is diagonal. This is in contrast to the XYZ-algorithm. In the XYZ-algorithm all β-matrices are diagonal, a subclass of quasi-monomial matrices. In that case it was shown in [24] that if there does not exist a set of signed permutations $\{\Pi_u\}$ such that $\Pi_u^\mathsf{T}\beta_{uv}\Pi_v$ is diagonal, then there also does not exist a set of orthogonal transformations $\{O_u\}$ such that $O_u^\mathsf{T}\beta_{uv}O_v$ is diagonal, and so it is sufficient to consider signed permutations.

This insight is somewhat surprising for the following reason. If one considers a single quasi-monomial matrix β, it holds that all other quasi-monomial matrices β' which can be obtained by orthogonal transformations $O_1^\mathsf{T}\beta O_2 = \beta'$ can also be obtained by signed permutations $\Pi_1^\mathsf{T}\beta\Pi_2 = \beta'$. One can see this by noting that the absolute values of the nonzero entries of β are its singular values, and so the singular value decomposition of β is related to all of the quasi-monomial matrices by shuffling and by flipping the signs of the rows and columns. The problem is coordinating these permutations.

Consider a matrix-weighted graph whose matrix weights are monomial matrices; in particular consider a triangle with three qubits and Hamiltonian of the form

$$H = H_{12} + H_{23} + H_{31}\ ,\ H_{uv} = X_uY_v + Y_uX_v.$$

The corresponding matrix weights of our graph are thus of the form

$$\beta_{uv} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

It is not hard to see that no permutations exist which simultaneously diagonalize all three matrices. However, if we apply the rotation

$$O = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{C.1}$$

at every vertex, then every matrix is diagonalized. Translating back to the application of basis changes, this transformation corresponds to applying T-gates, of the form $\left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{smallmatrix}\right)$, on all the qubits.

We see that in the case of quasi-monomial matrix-weighted graphs, the family of graphs which are equivalent under orthogonal transformations are not equivalent under signed permutations, and instead form sectors which depend on the graph topology. It is precisely the rank > 1 loops considered in our algorithm in section 6 which captures this nontrivial topological structure. In Algorithm 3, one needs to check the loop operators O_{P_e} (Algorithm 3, line 27)) in order to identify the rotation (C.1).

Appendix D. Heisenberg form under local unitaries is NP-hard. In this appendix we argue that deciding if a 2-local Hamiltonian can be put into *Heisenberg form* by single-qubit unitary rotations is NP-hard. Equivalently, for the case of a matrix-weighted graph of the kind considered in section 6, finding a set of orthogonal rotations $\{O_u\}$ that diagonalize every matrix weight is NP-hard.

DEFINITION D.1 (Heisenberg form). *A Hamiltonian of the form*

$$H = \sum_{\langle uv \rangle} J_x^{uv} X_u X_v + J_y^{uv} Y_u Y_v + J_z^{uv} Z_u Z_v$$

is said to be in Heisenberg form.

DEFINITION D.2 (cubic graph). *A cubic graph is a graph in which all vertices have degree three.*

DEFINITION D.3 (chromatic index). *The chromatic index of a graph is the minimum number of colors required to color the edges of the graph in such a way that no two adjacent edges have the same color.*

THEOREM D.4 (see [20]). *It is NP-complete to determine whether the chromatic index of a cubic graph is* 3 *or* 4.

COROLLARY D.5 (Heisenberg form is NP-hard). *It is NP-hard to determine if a* 2*-local Hamiltonian can be put into Heisenberg form by applying single qubit unitary rotations.*

Proof. The proof proceeds by reduction.

Consider a cubic graph $G = (E, V)$. Consider a two-local Hamiltonian H_G, acting on qubits associated with the vertices V. For every edge $e = (u, v) \in E$, H_G consists of a two-local term $P_u Q_v$, $P, Q \in \{X, Y, Z\}$, with the restriction that $\forall u \in V$, X_u, Y_u, and Z_u appear in exactly one such term. This last restriction is always possible because every vertex has degree 3.

If G has chromatic index 3, then H_G can be put into Heisenberg form by single-qubit unitaries. Let the coloring be given by the Pauli operators X, Y, and Z. For every edge $e = (u, v)$ colored by the Pauli operator R, map the Hamiltonian term $P_u Q_v$ to the term $R_u R_v$. Since every term in H_G acting on a given qubit acts with a different Pauli operator, and every edge incident on a vertex is given a different color, each Pauli operator acting on a given qubit is mapped to a unique Pauli operator, and so such a mapping can be given in terms of single-qubit Clifford operators.

For the reverse direction, we prove that if H_G can be put into Heisenberg form H'_G by single-qubit unitary rotations, then G has chromatic index 3. Putting H_G into Heisenberg form by single-qubit unitary rotations is equivalent to diagonalizing all β_{uv} matrices by orthogonal transformations, as defined in Proposition 4.2, for all edges (u, v). Consider that for a given edge (u, v), the matrix β_{uv} is rank-1. Thus for every edge (u, v), H'_G contains a single two-qubit term of the form $X_u X_v$, $Y_u Y_v$ or $Z_u Z_v$. Furthermore, all edges incident on a vertex must be associated with a different Pauli operator, since the transformation is unitary. Thus H'_G prescribes a 3 coloring, and G has chromatic index 3. ∎

Note that the above proof applies regardless of whether or not the Hamiltonians are restricted to being exactly two-local. Note also that problems of this type are ruled out by the No-Lone-YY condition introduced in section 6.

Acknowledgments. The authors would like to thank Sergey Bravyi and Daniel Lidar for their thoughtful comments on this work.

REFERENCES

[1] D. Aharonov and A. B. Grilo, *Stoquastic PCP vs. randomness*, in IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS), 2019, pp. 1000–1023, https://doi.org/10.1109/FOCS.2019.00065.

[2] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, *Adiabatic quantum computation is equivalent to standard quantum computation*, SIAM J. Comput., 37 (2007), pp. 166–194, https://doi.org/10.1137/S0097539705447323.

[3] T. Albash, *Role of nonstoquastic catalysts in quantum adiabatic optimization*, Phys. Rev. A, 99 (2019), 042334, https://doi.org/10.1103/PhysRevA.99.042334.

[4] T. Albash and D. A. Lidar, *Adiabatic quantum computation*, Rev. Modern Phys., 90 (2018), 015002, https://doi.org/10.1103/RevModPhys.90.015002.

[5] J. Bausch and E. Crosson, *Analysis and limitations of modified circuit-to-Hamiltonian constructions*, Quantum, 2 (2018), p. 94, https://doi.org/10.22331/q-2018-09-19-94.

[6] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, 1994, https://doi.org/10.1137/1.9781611971262.

[7] J. D. Biamonte and P. J. Love, *Realizable Hamiltonians for universal adiabatic quantum computers*, Phys. Rev. A, 78 (2008), 012352, https://doi.org/10.1103/PhysRevA.78.012352.

[8] R. F. Bishop and D. J. J. Farnell, *Marshall-Peierls sign rules, the quantum Monte Carlo method, and frustration*, in Recent Progress in Many-Body Theories, World Scientific, 2000, pp. 457–460, https://doi.org/10.1142/9789812792754_0052.

[9] S. Bravyi, A. J. Bessen, and B. M. Terhal, *Merlin-Arthur Games and Stoquastic Complexity*, preprint, http://arXiv.org/abs/quant-ph/0611021, 2006.

[10] S. Bravyi, D. P. Divincenzo, R. Oliveira, and B. M. Terhal, *The complexity of stoquastic local Hamiltonian problems*, Quantum Inf. Comput., 8 (2008), p. 361–385, https://dl.acm.org/doi/10.5555/2011772.2011773.

[11] S. Bravyi and B. Terhal, *Complexity of stoquastic frustration-free Hamiltonians*, SIAM J. Comput., 39 (2009), pp. 1462–1485, https://doi.org/10.1137/08072689X.

[12] J. Bringewatt, W. Dorland, S. P. Jordan, and A. Mink, *Diffusion Monte Carlo approach versus adiabatic computation for local Hamiltonians*, Phys. Rev. A, 97 (2018), 022323, https://doi.org/10.1103/PhysRevA.97.022323.

[13] S. Chandrasekharan, *Fermion bag approach to lattice field theories*, Phys. Rev. D, 82 (2010), 025007, https://doi.org/10.1103/PhysRevD.82.025007.

[14] T. Cubitt and A. Montanaro, *Complexity classification of local Hamiltonian problems*, SIAM J. Comput., 45 (2016), pp. 268–316, https://doi.org/10.1137/140998287.

[15] D. Hangleiter, I. Roth, D. Nagaj, and J. Eisert, *Easing the Monte Carlo sign problem*, Sci. Adv., 6 (2020), eabb8341, https://doi.org/10.1126/sciadv.abb8341.

[16] M. B. Hastings, *Obstructions to classically simulating the quantum adiabatic algorithm*, Quantum Inf. Comput., 13 (2013), p. 1038–1076, https://dl.acm.org/doi/10.5555/2535639.2535647.

[17] M. B. Hastings, *How quantum are non-negative wavefunctions?*, J. Math. Phys., 57 (2016), 015210, https://doi.org/10.1063/1.4936216.

[18] N. Hatano and M. Suzuki, *Representation basis in quantum Monte Carlo calculations and the negative-sign problem*, Phys. Lett. A, 163 (1992), pp. 246–249, https://doi.org/10.1016/0375-9601(92)91006-D.

[19] I. Hen, *Resolution of the sign problem for a frustrated triplet of spins*, Phys. Rev. E, 99 (2019), 033306, https://doi.org/10.1103/PhysRevE.99.033306.

[20] I. Holyer, *The NP-completeness of edge-coloring*, SIAM J. Comput., 10 (1981), pp. 718–720, https://doi.org/10.1137/0210055.

[21] L. Hormozi, E. W. Brown, G. Carleo, and M. Troyer, *Nonstoquastic Hamiltonians and quantum annealing of an Ising spin glass*, Phys. Rev. B, 95 (2017), 184416, https://doi.org/10.1103/PhysRevB.95.184416.

[22] M. Ioannou, *Stoquastic Hamiltonians and the Monte Carlo sign problem*, Ph.D thesis, LMU Munich, 2019; supervision by Prof. B. M. Terhal.

[23] M. Jarret, S. P. Jordan, and B. Lackey, *Adiabatic optimization versus diffusion Monte Carlo methods*, Phys. Rev. A, 94 (2016), 042318, https://doi.org/10.1103/PhysRevA.94.042318.

[24] J. Klassen and B. M. Terhal, *Two-local qubit Hamiltonians: When are they stoquastic?*, Quantum, 3 (2019), p. 139, https://doi.org/10.22331/q-2019-05-06-139.

[25] D. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, 2005.

[26] E. Y. Loh, J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L.

Sugar, *Sign problem in the numerical simulation of many-electron systems*, Phys. Rev. B, 41 (1990), pp. 9301–9307, https://doi.org/10.1103/PhysRevB.41.9301.

[27] M. Marvian, D. A. Lidar, and I. Hen, *On the computational complexity of curing non-stoquastic Hamiltonians*, Nature Commun., (2019), https://www.nature.com/articles/s41467-019-09501-6.

[28] H. Nishimori and K. Takada, *Exponential enhancement of the efficiency of quantum annealing by non-stoquastic Hamiltonians*, Frontiers ICT, 4 (2017), 2, https://doi.org/10.3389/fict.2017.00002.

[29] I. Ozfidan et al., *Demonstration of a nonstoquastic Hamiltonian in coupled superconducting flux qubits*, Phys. Rev. Appl., 13 (2020), 034037, https://doi.org/10.1103/PhysRevApplied.13.034037.

[30] Z. Ringel and D. L. Kovrizhin, *Quantized gravitational responses, the sign problem, and quantum complexity*, Sci. Adv., 3 (2017), e1701758, https://doi.org/10.1126/sciadv.1701758.

[31] M. Suzuki, *Quantum Monte Carlo Methods in Condensed Matter Physics*, World Scientific, 1993, https://doi.org/10.1142/2262.

[32] G. Torlai, J. Carrasquilla, M. T. Fishman, R. G. Melko, and M. P. A. Fisher, *Wavefunction positivization via automatic differentiation*, Phys. Rev. Res., 2 (2020), 032060, https://doi.org/10.1103/PhysRevResearch.2.032060.

[33] M. Troyer, *Boulder School* 2010*: Computational and Conceptual Approaches to Quantum Many-Body Systems*, Lecture notes, Yale University, 2010.

[34] W. Vinci and D. A. Lidar, *Non-stoquastic Hamiltonians in quantum annealing via geometric phases*, npj Quant. Inf., 3 (2017), 38, https://www.nature.com/articles/s41534-017-0037-z.

[35] P. Werner, A. Comanac, L. de' Medici, M. Troyer, and A. J. Millis, *Continuous-time solver for quantum impurity models*, Phys. Rev. Lett., 97 (2006), 076405, https://doi.org/10.1103/PhysRevLett.97.076405.