Sum-of-Squares inspired Quantum Metaheuristic for Polynomial Optimization with the Hadamard Test and Approximate Amplitude Constraints

Iria W. Wang,[1] Robin Brown,[2] Taylor L. Patti,[3] Anima Anandkumar,[4] Marco Pavone,[2,3] and Susanne F. Yelin[1]

[1] *Department of Physics, Harvard University, Cambridge, MA 02138, USA*
[2] *Autonomous Systems Laboratory, Stanford University, Stanford, CA 94305, USA*
[3] *NVIDIA, Santa Clara, California 95051, USA*
[4] *Department of Computing + Mathematical Sciences (CMS), California Institute of Technology (Caltech), Pasadena, CA 91125, USA*

(Dated: August 16, 2024)

Quantum computation shows promise for addressing numerous classically intractable problems, such as optimization tasks. Many optimization problems are NP-hard, meaning that they scale exponentially with problem size and thus cannot be addressed at scale by traditional computing paradigms. The recently proposed quantum algorithm [1] addresses this challenge for some NP-hard problems, and is based on classical semidefinite programming (SDP). In this manuscript, we generalize the SDP-inspired quantum algorithm to sum-of-squares programming, which targets a broader problem set. Our proposed algorithm addresses degree-k polynomial optimization problems with $N \leq 2^n$ variables (which are representative of many NP-hard problems) using $O(nk)$ qubits, $O(k)$ quantum measurements, and $O(\text{poly}(n))$ classical calculations. We apply the proposed algorithm to the prototypical Max-kSAT problem and compare its performance against classical sum-of-squares, state-of-the-art heuristic solvers, and random guessing. Simulations show that the performance of our algorithm surpasses that of classical sum-of-squares after rounding. Our results further demonstrate that our algorithm is suitable for large problems and approximates the best known classical heuristics, while also providing a more generalizable approach compared to problem-specific heuristics.

I. INTRODUCTION

Recent research in quantum computation has suggested that quantum resources may enable computational advantages. Such an advantage could have profound implications in several fields, including drug development, materials science, machine learning, and operations research [2–5]. An application of particular interest is combinatorial optimization. While combinatorial optimization problems are ubiquitous [6–9], they are often NP-hard, meaning that their complexity scales exponentially with problem size. This rapidly renders them intractable for classical computers. Since this exponential scaling is similar to that of the quantum Hilbert space, it has inspired numerous quantum investigations [1, 10–23]. However, exactly solving NP-hard optimization problems on variational quantum devices maintains many of the same exponential overheads as classical solvers [24].

To combat these overheads, more efficient yet less accurate techniques known as *approximation algorithms* are typically used. Improving the performance and study of these algorithms remains a central research area in classical optimization theory [6–8, 25–33]. One class of classical approximation algorithms that have garnered significant attention in quantum computing research is semidefinite programming (SDP). SDP targets quadratic optimization and is often used in the context of NP-hard problems, e.g. MaxCut and Max-2SAT, which are expressible as quadratic optimizations over integer-valued variables [28]. SDPs can be used to construct *relaxations*, as they are optimized over non-integer values – those relaxed solutions can then be rounded back into the space of feasible solutions. As polynomial-time algorithms, SDPs work well for problem instances of considerable size, but ultimately become computationally impractical at scales relevant to many industrial and research applications.

Significant work has been done to develop a quantum algorithm that efficiently approximates solutions to Max-2SAT and similar NP-hard problems. Quantum semidefinite programs (QSDPs) have been proposed [10–13, 15, 16], but require the estimation of up to $O(2^n)$ observables for some problems, and many are unsuitable for near-term devices. Variational approaches [14, 17–22] are somewhat more suitable for near-term devices, but continue to require an exponential number of observables in the worst case. To address these limitations, quantum semidefinite programming with the Hadamard test and approximate amplitude constraints (HTAAC-QSDP) [1] was recently proposed as a nearer-term and sample-efficient variational QSDP. HTAAC-QSDP can find approximate solutions for problems with up to $N \leq 2^n$ variables using only $O(n) = O(\log_2 N)$ qubits, $O(n)$ expectation values, and $O(\text{poly}(n))$ classical calculations[1]. This efficiency is largely furnished by estimating the objective functions with the Hadamard test and enforcing the problem constraints with approximate amplitude constraints.

Like classical SDPs, HTAAC-QSDP specifically ad-

[1] Throughout this manuscript, we use N to denote the number of variables in the problem and $n \geq \log_2 N$ to denote the number of qubits required to encode N variables.

dresses quadratic optimization. As an extension of HTAAC-QSDP towards higher-dimensional tasks (e.g., NP-hard problems that are represented as higher-degree polynomial optimization problems), we propose and simulate a quantum algorithm based on a well-known classical approach, sum-of-squares (SOS). Our proposed algorithm, an SOS-inspired quantum metaheuristic for polynomial optimization with the Hadamard test and approximate amplitude constraints (HTAAC-QSOS), targets degree-k polynomial optimization problems. HTAAC-QSOS uses $O(nk)$ qubits, $O(k)$ quantum measurements, and $O(\text{poly}(n))$ classical calculations to approximate solutions for problems with $N \leq 2^n$ variables. We focus on formulating the algorithm for an illustrative NP-hard problem that is expressible as degree-k polynomial optimization: Max-kSAT.

In this manuscript, we present the general HTAAC-QSOS framework and apply it to the prototypical Max-kSAT problem, comparing its performance against both classical SOS and state-of-the-art heuristic solvers. Simulations of HTAAC-QSOS are run for Max-3SAT instances with up to 110 variables and 1100 clauses. Our results demonstrate that HTAAC-QSOS is tractable for large problems and is competitive with the best known classical solutions. Section II describes the pre-existing approximation methods (namely, SDP, HTAAC-QSDP, and SOS) in the context of Max-kSAT. Section III introduces the proposed HTAAC-QSOS. We compare HTAAC-QSOS to classical benchmarks by simulating HTAAC-QSOS for Max-3SAT on a classical machine. Section IV discusses the results and suggests a future research direction.

II. PRELIMARIES

In this section we review the existing approximation techniques relevant to our proposed method, HTAAC-QSOS, in the context of Max-kSAT. This includes the two standard classical optimization techniques – semidefinite programming (SDP) and sum-of-squares (SOS) programming – which serve as both algorithmic foundations and practical benchmarks for HTAAC-QSOS. We also summarize the recently-proposed HTAAC-QSDP [1], upon which HTAAC-QSOS is based. We begin by introducing Max-kSAT.

A. The Max-kSAT Problem

Maximum Satisfiability (Max-SAT) is a quintessential example of an NP-hard combinatorial optimization problem. Max-SAT can be used in practical settings to solve optimization problems arising in data analysis and machine learning, and studying this problem provides insight into computational complexity theory and analysis of approximation algorithms [7, 25]. Given a list of Boolean (true or false) clauses each consisting of Boolean variables, the goal of Max-SAT is to find a truth assignment of these variables that maximizes the number of satisfied clauses (i.e., those that evaluate to *True*). Max-kSAT is a special case of Max-SAT, where each clause is restricted to contain k variables. Max-kSAT is NP-hard, simpler to analyze, generalizable to Max-SAT, and well-studied [7, 25, 26, 28, 29], making it an attractive benchmark for quantum computing approaches to NP-hard problems. Max-2SAT ($k = 2$) can be expressed as a quadratic optimization problem. Generally, Max-kSAT can be formulated as a degree-k polynomial optimization problem over integer-valued variables.

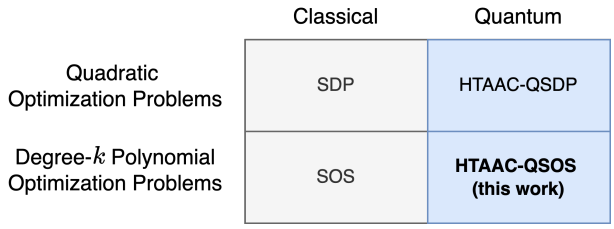

FIG. 1: **Algorithmic Context for HTAAC-QSOS.** In classical optimization theory, the standard approach towards quadratic optimization problems is SDP (semidefinite programming). The generalization of this approach for degree-k polynomial optimization problems, where k is a natural number, is called SOS (sum-of-squares) programming. In the quantum regime, the recently-proposed HTAAC-QSDP (SDP-inspired quantum metaheuristic for quadratic optimization) [1] is a variational quantum analog to classical SDPs. Our proposed HTAAC-QSOS (SOS-inspired quantum metaheuristic for polynomial optimization) is likewise a higher-dimensional generalization of HTAAC-QSDP.

SDP for Max-2SAT and SOS programming for Max-kSAT are standard classical relaxations that upper bound the solution. Figure 1 contextualizes these approximation algorithms. In addition, heuristic solvers (often based on local search or variational algorithms) aim to find feasible solutions to optimization problems quickly without guarantees on solution quality [25, 26, 29]. Since industrial applications often prioritize speed, these solvers may work well in practice but are frequently catered towards specific problems.

For concreteness, we focus our discussions of SDPs and HTAAC-QSDP on the problem Max-2SAT. We then describe how similar methodology may be extended to solve the Max-kSAT via SOS. Despite our focus on SAT problems, we remind the reader that these optimization techniques readily generalize to other optimization problems [9, 34].

B. Max-2SAT and Semidefinite Programming

Max-2SAT presents a set of clauses, each containing two Boolean variables. The goal is to maximize the

number of satisfiable clauses by optimally assigning the Boolean variables.

In this section, we detail the Goemans-Williamson (GW) approximation algorithm for Max-2SAT [28]. The GW algorithm is highly regarded due its favorable approximation ratio, which is conjectured to be optimal among polynomial-time algorithms. In particular, it is proven that the GW algorithm for Max-2SAT will deliver solutions with an expected value of at least 0.87856 times the optimal number of satisfied clauses. To approximate a solution using the GW algorithm, we first express Max-2SAT as a quadratic optimization problem. Then, we construct a relaxation of the problem as a semidefinite program (SDP). Finally, the solution to this SDP is rounded to produce a feasible, albeit potentially suboptimal, solution to the Max-2SAT problem.

1. *Max-2SAT*

In this section we consider a Max-2SAT instance on $N-1$ boolean variables, whose values are represented as $\{x_i\}_{i\in\{1,\dots,N-1\}}$[2]. A Max-2SAT instance is a series of clauses written, without loss of generality, in Conjunctive Normal Form (CNF)[3]. In CNF, a clause containing variables x_i and x_j is written as $(x_i \vee x_j)$, where either or both of x_i and x_j may be replaced by their negations (denoted $\neg x_i$ and $\neg x_j$, respectively) and $\vee$ denotes the logical "or". Max-2SAT is the sum of all these clauses. We note that Max-kSAT problems are similarly expressed, with k variables per clause.

We define another set of N variables that correspond to the boolean variables $\{x_i\}_{i\in\{1,\dots,N-1\}}$, writing $\{y_i\}_{i\in\{0,1,\dots,N-1\}}$ such that $y_i \in \{1,-1\}$ for all $i \in \{0,1,\dots,N-1\}$. Notice that we have introduced the variable y_0 which has no corresponding boolean variable. This y_0 defines the truth value, that is, we say x_i is true if $y_i = y_0$ and false if $y_i \neq y_0$. While introducing y_0 is not necessary for a classical representation of the problem, we will see its importance for the quantum formulation later in this section.

To denote the truth value of a clause C, we write $v(C) = 1$ if C is true and $v(C) = 0$ if C is false. Then, the truth value of x_i and $\neg x_i$ may be expressed as a function of $\{y_i\}_{i\in\{0,1,\dots,N-1\}}$:

$$v(x_i) = \frac{1+y_0y_i}{2} \text{ and } v(\neg x_i) = \frac{1-y_0y_i}{2}. \tag{1}$$

The truth value of a clause $(x_i \vee x_j)$ is then given by

$$\begin{aligned} v(x_i \vee x_j) &= 1 - v(\neg x_i)v(\neg x_j) \\ &= \frac{1+y_0y_i}{4} + \frac{1+y_0y_j}{4} + \frac{1-y_iy_j}{4}. \end{aligned} \tag{2}$$

The value of other clauses can be similarly expressed. If x_i or x_j are negated, then we replace the corresponding y_i or y_j with $-y_i$ or $-y_j$. The truth value of all clauses of length-2 may then be expressed as a quadratic function. The maximum number of satisfiable clauses is therefore written as a maximization over $\sum_\ell v(C_\ell)$ where C_ℓ is the ℓth clause in the given list of clauses. This is simply a linear combination of the quadratic terms seen in Eq. (2), yielding

$$\begin{aligned} \text{maximize} \quad & \sum_{i<j}(a_{i,j}(1-y_iy_j) + b_{i,j}(1+y_iy_j))) \\ \text{subject to} \quad & y_i \in \{-1,1\}, \forall i \in \{0,\dots,N-1\}, \end{aligned} \tag{3}$$

where $a_{i,j}$ and $b_{i,j}$ are scalar constants defined by the specific problem instance, and $i,j \in \{0,1,\dots,N-1\}$. While y_0 has a separate definition from other variables in $\{y_i\}_{i\in\{1,\dots,N-1\}}$, we can treat y_0 the same as $\{y_i\}_{i\in\{1,\dots,N-1\}}$ once this quadratic function is determined.

In classical optimization, Eq. (3) is called a quadratic optimization problem. Other NP-hard problems such as MaxCut and MaxBisection may be similarly expressed as quadratic optimization problems. The conversion from the NP-hard problem instance to a quadratic optimization problem is done in polynomial time [9, 28].

We note that the polynomial objective function consists of only constant and quadratic (i.e., even-degree) terms. If we were to eliminate y_0 (for example, by fixing the convention that $y_i = 1$ indicates x_i as true and $y_i = -1$ indicates x_i as false for all $i \in \{1,\dots,N-1\}$), then the resulting polynomial objective would also contain linear (odd-degree) terms. We find even-degree optimization problems favorable for their matrix forms, that is, their variable matrices can be defined as an outer product between vectors of variables. The role of y_0 is thus to formulate the polynomial objective with only even-degree terms. In general, any polynomial can be made even by introducing a variable analogous to y_0.

2. *Semidefinite Programming (SDP)*

Semidefinite programming (SDP) and corresponding rounding procedures are standard techniques in classical optimization for bounding quadratic optimization problems such as Max-2SAT.

SDP is a class of convex optimization problems that aims to extremize a linear objective function over symmetric positive semidefinite matrices $\mathbb{S}^+$ under a set of constraints. With diverse applications in control theory and combinatorial optimization [8, 27, 32, 33, 36], SDPs

[2] We note that $N-1$ is a deliberate choice to simplify notation when we introduce an auxiliary variable.

[3] In Conjunctive Normal Form, clauses consist of literals (boolean variables and their negations) connected by the logical "or", denoted by $\vee$. These clauses are sometimes connected by the logical "and", denoted by $\wedge$. Any propositional formula, or equation with some truth value, may be written in CNF form [35].

are often used to bound solutions to NP-hard maximization problems. Specifically, such NP-hard problems are expressed as quadratic optimizations (often over integer-valued variables), the encoding of which is relaxed such that optimization can take place over the space of positive semidefinite matrices. This new SDP matrix formulation provides a more tractable and relaxed form of the original problem; however, we emphasize that this relaxed form is not equivalent to the original NP-hard problem. Rather, the SDP relaxation strictly broadens the solution space, upper bounding the "feasible region" of the original NP-hard problem. The solution that corresponds to this upper bound is potentially outside the feasible region, so a rounding procedure returns it to the valid solution space. This yields an approximate lower bound (but possibly sub-optimal) solution.

In general, the relaxed SDP form of these optimization problems is given as

$$\begin{aligned} &\underset{X \in \mathbb{S}^+}{\text{minimize}} && \langle W, X\rangle \\ &\text{subject to} && \langle A_\mu, X\rangle = b_\mu \, \forall \mu \leq M, \end{aligned} \tag{4}$$

where W is an $N \times N$ symmetric matrix encoding a specific problem instance, matrix A_μ and scalar b_μ describe problem constraints, and the angled brackets denote the matrix outer product

$$\langle A, B\rangle = \mathrm{Tr}\big(A^T B\big) = \sum_{i,j=1}^{N} A_{i,j} B_{i,j} \tag{5}$$

for some matrices A and B. These SDPs are exactly solvable using classical techniques such as interior point methods [8].

Returning to the Max-2SAT problem, we can reformulate the quadratic optimization form of Max-2SAT in Eq. (3) as an SDP. The variables y_i are relaxed to take on a vector form $v_i \in S_N$, where $S_N \subset \mathbb{R}^N$ represents the unit sphere in N dimensions. Then, the quadratic form in Eq. (3) is relaxed to

$$\begin{aligned} &\text{maximize} && \sum_{i<j} (W^+_{i,j} - v_i^T v_j W^-_{i,j}) \\ &\text{subject to} && v_i \in S_N, \quad \forall i \in \{0, ..., N-1\}, \end{aligned} \tag{6}$$

where $W^+_{i,j} \equiv a_{i,j} + b_{i,j}$ and $W^-_{i,j} \equiv a_{i,j} - b_{i,j}$ are matrix elements. The objective in Eq. (6), or number of satisfiable clauses, becomes

$$\frac{1}{2}\left(\langle W^+, \mathbf{1}_N\rangle + \langle W^-, X\rangle\right) \tag{7}$$

where $\mathbf{1}_N$ is an $N \times N$ matrix of all ones, and $X_{i,j} = v_i^T v_j$. Eq. (6) is equivalent to the SDP

$$\begin{aligned} &\underset{X \in \mathbb{S}^+}{\text{minimize}} && \langle W^-, X\rangle \\ &\text{subject to} && X_{i,i} = 1, \quad \forall i \leq N. \end{aligned} \tag{8}$$

We remark again that the SDP formulation is a relaxation of Max-2SAT. To obtain an approximate solution within the feasible space of the Max-2SAT instance, un-rounded solutions $v_i \in S_N$ are rounded back to $y_i \in \{1, -1\}$ by determining which side of a random hyperplane they fall on. The resulting $\{y_i\}_{i \in \{0,1,...,N-1\}}$ are substituted into Eq. (3) to determine the number of satisfied clauses in the rounded solution.

C. HTAAC-QSDP for Max-2SAT

In this section, we review the recently-proposed HTAAC-QSDP [1]. As a variational quantum analog to SDPs, HTAAC-QSDP finds heuristic solutions for problems with $N \leq 2^n$ variables using only $O(n)$ qubits, $O(1)$ quantum measurements, and $O(\mathrm{poly}(N))$ classical calculations. As with other variational quantum algorithms [23], the key elements of HTAAC-QSDP lie in the formulation of the loss function. The loss is determined using observables on the quantum state $|\psi\rangle = U_V |0\rangle^{\otimes n}$, and is minimized via the variational quantum circuit U_V. Although HTAAC-QSDP is a general SDP technique, the precise loss function formulation is problem-dependent. In the following sections, we detail the application to Max-2SAT.

1. *Evaluating the Objective*

Beginning with the SDP formulation in Eq. (8), we replace the matrix X with the density matrix $\rho = |\psi\rangle\langle\psi|$, where $|\psi\rangle = \begin{pmatrix}\psi_0 & \psi_1 & ... & \psi_{2^n-1}\end{pmatrix}^T$ in the computational basis. In particular, ψ_i corresponds to y_i for all $i \in \{0, 1, ..., N-1\}$, where we recall $N \leq 2^n$. The elements $\rho_{i,j}$ of ρ represent $y_i y_j$, such that the quantum equivalent of Eq. (8) is [4]

$$\begin{aligned} &\underset{\rho}{\text{minimize}} && \langle W^-, \rho\rangle \\ &\text{subject to} && \rho_{i,i} = 2^{-n}, \quad \forall i \leq N. \end{aligned} \tag{9}$$

The objective function may be written as as an expectation value

$$\langle W^-, \rho\rangle = \mathrm{Tr}\big(W^- \rho\big) = \langle\Psi| W^- |\Psi\rangle \tag{10}$$

and W^- may be used to generate a unitary U_{W^-},

$$\begin{aligned} U_{W^-} &= e^{i\alpha W^-} \\ &= 1 + i\alpha W^- - \frac{\alpha^2}{2!}(W^-)^2 + O((W^-)^3). \end{aligned} \tag{11}$$

[4] To ensure that the dimensions of W^- and ρ match, W^- may be "padded" with $2^n - N$ rows and $2^n - N$ columns of zeroes. In this work, the notation is the same for padded and non-padded versions of a matrix.

Then, the value of the objective function $\langle\psi| W^- |\psi\rangle$ may be approximated by using a Hadamard test to evaluate

$$\mathrm{Im}(\langle\psi| U_{W^-} |\psi\rangle) \approx \alpha \langle\psi| W^- |\psi\rangle . \tag{12}$$

The Hadamard test is a quantum computing subroutine that estimates $\mathrm{Im}(\langle\psi| U_{W^-} |\psi\rangle)$ and other expectation values in the loss function. To generalize this subroutine we could simply replace $|\psi\rangle$ with an arbitrary n-qubit state and U_{W^-} with an arbitrary n-qubit unitary. The circuit used to execute the Hadamard test is shown in Figure 3, which shows that the Hadamard test uses only a single expectation value on an ancilla qubit

$$\langle\sigma^z_{\text{anc.}}\rangle_{W^-} \equiv \mathrm{Im}(\langle\psi| U_{W^-} |\psi\rangle) \tag{13}$$

instead of the $N \leq 2^n$ expectation values that would be otherwise required to fully characterize ρ.

We note that the variational circuit U_V is designed such that the elements ψ_i of $|\psi\rangle$ must be real. Thus, the objective $\langle\Psi| W^- |\Psi\rangle$ is real, and Eq. (12) holds. Further discussion on the limits of this approximation is included in the original HTAAC-QSDP manuscript [1].

2. *Approximate Amplitude Constraints*

For Max-2SAT we wish to approximately enforce the constraint $\rho_{i,i} = 2^{-n}$ from Eq. (9). We note that enforcing $\rho_{i,i} = 2^{-n}$ is equivalent to enforcing $|\psi_0|^2 = |\psi_1|^2 = ... = |\psi_{2^n-1}|^2$ up to normalization. This yields $2^n - 1$ degrees of freedom, defined by $2^n - 1$ unique equations[5]. We can rewrite this set of $2^n - 1$ equations using the following Pauli strings of length 1 to n:

$$\begin{aligned}
&\langle\sigma^z_a, \rho\rangle = 0,\ \forall a \leq n, \text{ yielding } \binom{n}{1} \text{ equations} \\
&\langle\sigma^z_a\sigma^z_b, \rho\rangle = 0,\ \forall a < b \leq n, \text{ yielding } \binom{n}{2} \text{ equations} \\
&... \\
&\langle\sigma^z_1\sigma^z_2...\sigma^z_n, \rho\rangle \text{ , yielding } \binom{n}{n} \text{ equations.}
\end{aligned} \tag{14}$$

The total number of equations is given by

$$\sum_{p=1}^{n} \binom{n}{p} = \sum_{p=0}^{n} \binom{n}{p} - 1 = 2^n - 1. \tag{15}$$

Each of the Pauli strings in Eq. (14) yields an equation $\sum_i c_i |\psi_i|^2 = 0$, for $c_i \in \{1, -1\}$.

[5] A unique set of equations is defined here as a set in which not equation can be deduced from other equations in the set. For example, $\{a = b, b = c\}$ are unique, but $\{a = b, b = c, a = c\}$ are not unique.

From Eq. (15), exactly enforcing the constraint requires $O(2^n)$ observables. To avoid this, Patti et al. [1] propose *approximate amplitude constraints*. We approximately enforce $\rho_{i,i} = 2^{-n}$ by truncating Eq. (15) at $p = 2$, such that we only consider the $O(n^2)$ Pauli strings up to length 2. All other terms give an extra degree of freedom (d.o.f.). This series may be truncated at higher p to make the approximation more precise, in exchange for less-favorable polynomial scaling. The effect of this truncation is further explored in the original HTAAC-QSDP paper [1].

Since this method is approximate, HTAAC-QSDP includes an additional component to supplement the above Pauli string constraints. We introduce a diagonal matrix P and a "population-balancing" unitary

$$\begin{aligned}
&U_P = e^{i\beta P} \\
&\text{where } P_{i,i} = -\left(P_{\max} - \sum_j \left|W^-_{i,j}\right|\right), \\
&P_{\max} = \max_i \sum_j W^-_{i,j}.
\end{aligned} \tag{16}$$

This helps to offset any asymmetric weights, and the expectation value of this unitary may be estimated using a Hadamard test.

3. *The Loss Function*

All three elements – the objective function, population-balancing unitary, and Pauli strings – may be combined as a single loss function to be minimized. The loss function is given by

$$\mathcal{L} = \langle\sigma^z_{\text{anc.}}\rangle_{W^-} + \lambda \left(\sum_{\langle\sigma,\rho\rangle\in\mathcal{P}} \langle\sigma, \rho\rangle + \langle\sigma^z_{\text{anc.}}\rangle_P\right). \tag{17}$$

The first term $\langle\sigma^z_{\text{anc.}}\rangle_{W^-}$ results from the Hadamard test used to estimate the objective function, the second term $\sum_{\langle\sigma,\rho\rangle\in\mathcal{P}} \langle\sigma, \rho\rangle$ represents the set of Pauli strings $\mathcal{P}$, and the third term $\langle\sigma^z_{\text{anc.}}\rangle_P$ is from the Hadamard test used to estimate the population-balancing unitary. The second and third terms make up our approximate amplitude constraints, and are weighed by a Lagrange multiplier λ.

4. *Retrieving Solutions*

Once the loss is minimized, there are two ways to retrieve solutions from the resulting quantum state. One method uses sample-efficent estimation (SEE) and yields an un-rounded solution. Another method requires full state tomography (FST), but its complexity does not depend on the degree of the polynomial and the result mimics classical rounded solutions.

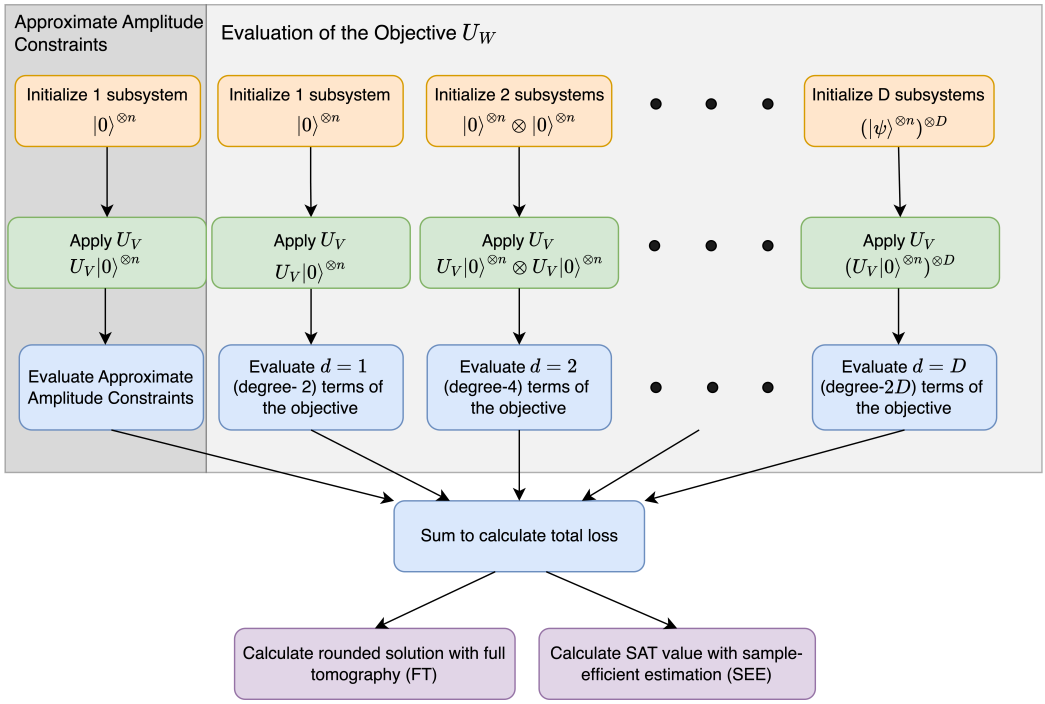

FIG. 2: **Our proposed HTAAC-QSOS.** HTAAC-QSOS is a variational algorithm, such that a n-qubit variational circuit U_V is applied to quantum subsystems (green) and iteratively modified to minimize a specified loss function. The loss approximates a degree-$2D$ polynomial optimization problem with up to $N \leq 2^n$ variables using expectation values calculated from the total quantum system (blue). The solution is extracted by conducting full state tomography (FST) and rounding the resulting quantum state, or by using sample-efficient estimation (SEE) to directly estimate the un-rounded SAT value (purple).

SEE may be used to directly estimate a solution from $|\psi\rangle$ with the quantum analog to Eq. (7),

$$2^{n-1}\left(\langle 0|^{\otimes n} H^{\otimes n} W^{+} H^{\otimes n} |0\rangle^{\otimes n} + \langle\psi| W^{-} |\psi\rangle\right). \quad (18)$$

The first term represents constant terms, where $H^{\otimes n}|0\rangle^{\otimes n}$ is the equal superposition of all states. This may be evaluated using the Hadamard test, where the unitary is $U_{W^+} \equiv e^{i\alpha W^+}$ acting on the quantum state $H^{\otimes n}|\psi\rangle^{\otimes n}$. We evaluate

$$\begin{aligned}\langle\sigma^z_{\text{anc.}}\rangle^{\otimes H}_{W^+} &\equiv \text{Im}[\langle 0|^{\otimes n} H^{\otimes n} U_{W^+} H^{\otimes n} |0\rangle^{\otimes n}] \\ &\approx \alpha \langle 0|^{\otimes n} H^{\otimes n} W^{+} H^{\otimes n} |0\rangle^{\otimes n}.\end{aligned} \quad (19)$$

The second term is equivalent to quadratic terms in the objective. Thus, the solution given by SEE is

$$\frac{2^{n-1}}{\alpha}\left(\langle\sigma^z_{\text{anc.}}\rangle^{\otimes H}_{W^+} - \langle\sigma^z_{\text{anc.}}\rangle_{W^-}\right). \quad (20)$$

While this solution is un-rounded, it is a good approximation of a rounded solution in the limit of well-behaved constraints. When the constraints are exactly enforced, the solution approaches the optimal rounded solution.

Alternatively, by using FST and a simple rounding procedure, we may extract a guaranteed feasible solution. We assign $y_i = \text{sign}(\psi_i)$ and substitute the resulting $\{y_i\}_{i\in\{0,1,\ldots,N-1\}}$ into the objective from Eq. (3). While this is sample-intensive on a quantum machine, it may be suitable as a quantum-inspired algorithm [37].

D. Max-kSAT and Sum-of-Squares Programming

In this section, we describe how Max-kSAT may be expressed as a polynomial optimization problem and approximated with classical sum-of-squares (SOS) programming. We first consider an extension from Max-2SAT to Max-3SAT.

1. *The Polynomial Optimization Problem*

In Max-3SAT, clauses are written in the form $(x_i \vee x_j \vee x_k)$, where any of x_i, x_j, and x_k may be replaced by their negations. By incorporating y_0 analogously to Max-2SAT, the truth value of the clause $(x_i \vee x_j \vee x_k)$ is expressed as

$$\begin{aligned}v(x_i \vee x_j \vee x_k) &= 1 - v(\neg x_i)v(\neg x_j)v(\neg x_k) \\ &= \frac{1+y_0y_i}{8} + \frac{1+y_0y_j}{8} + \frac{1+y_0y_k}{8} + \frac{1-y_iy_j}{8} \\ &\quad + \frac{1-y_iy_k}{8} + \frac{1-y_jy_k}{8} + \frac{1+y_0y_iy_jy_k}{8}.\end{aligned} \quad (21)$$

The value of other clauses can be similarly expressed. If any of x_i, x_j, and x_k are negated, then we replace the corresponding y_i, y_j, or y_k with $-y_i$, $-y_j$, or $-y_k$. Once these polynomials are formulated for every clause, y_0 may be treated identically to $\{y_i\}_{i\in\{1,\ldots,N-1\}}$. The truth value of all clauses of length-3 may then be expressed as a degree-4 polynomial. The maximum number of satisfiable clauses is therefore written as a maximization over $\sum_\ell v(C_\ell)$, where C_ℓ is the ℓth clause in the given list of clauses. This is a linear combination of the quadratic and degree-4 terms in Eq. (21),

$$\sum_\ell v(C_\ell) = \sum_{i<j}\left(a^{(1)}_{i,j}(1-y_iy_j) + b^{(1)}_{i,j}(1+y_iy_j)\right) + \sum_{i<j<q<l}\left(a^{(2)}_{ij,ql}(1-y_iy_jy_qy_l) + b^{(2)}_{ij,ql}(1+y_iy_jy_qy_l)\right) \tag{22}$$

where $a^{(1)}_{i,j}$, $b^{(1)}_{i,j}$, $a^{(2)}_{ij,ql}$, and $b^{(2)}_{ij,ql}$ are scalar constants defined by the specific problem instance, and $i,j,q,l \in \{0,1,\ldots,N-1\}$. The result is a non-convex degree-4 polynomial objective, consisting of only even-degree terms.

We note that we may obtain a degree-3 polynomial if we eliminate y_0 (for example, by fixing the convention that $y_i = 1$ indicates x_i as true and $y_i = -1$ indicates x_i as false for all $i \in \{1,\ldots,N-1\}$). Classical SOS can target degree-3 polynomials, but formulating polynomials with even-degree terms using y_0 is more suitable to our quantum approach.

To generalize this formulation to Max-kSAT, we introduce some notation. Max-kSAT may be formulated as a degree-k polynomial without y_0, and as a degree-$2D$ polynomial with y_0 where $D \equiv \text{ceil}(k/2)$. The maximum degree of the polynomial objective is $2D$, and we also we define $d \in \{1,\ldots,D\}$ such that $2d$ denotes the degree of individual terms in the objective. For example, in Max-3SAT we have a degree-4 polynomial objective, and therefore $D = 2$. Within this degree-4 polynomial we have constant, degree-2, and degree-4 terms, such that $2d = 0$, 2, and 4 respectively. Because constant terms do not impact the solution of an optimization problem, they are dropped and $d \in \{1,2\}$. Since any polynomial can be made to have only even-degree terms using y_0 and our proposed algorithm makes use of the even-degree characteristic, we focus on degree-$2D$ polynomials in our discussions.

2. *Sum-of-Squares (SOS)*

SOS programming is a core discipline in operations research [30]. In this section, we provide a cursory overview of the relevant SOS formulae, providing a more detailed treatment in the Appendix.

SOS programming determines whether a degree-$2D$ polynomial function $F(y)$, where $y = \{y_i\}_{i\in\{0,1,\ldots,N-1\}}$, can be factored as a sum of squares. We express $F(y)$ as

$$F(y) = b^T Q b, \tag{23}$$

where Q is a symmetric matrix and

$$b^T = \begin{pmatrix}1 & y_1 & y_2 & \ldots & y_n & y_1y_2 & \ldots & y_n^D\end{pmatrix}. \tag{24}$$

is a "polynomial basis". This basis enables the generalization to higher-degree polynomials. The matrix Q can be interpreted as a "weight" matrix of coefficients. It is proven that $F(y) = b^TQb$ is a sum of squares if and only if Q is positive semidefinite. SOS is therefore solvable by SDP [30].

In the context of polynomial optimization, we can define $F(y) \equiv -p(y) - \gamma$ where $p(y)$ is the polynomial objective and γ is a real number. Writing $F(y)$ as SOS is an algebraic certificate of nonnegativity, that is, $F(y) > 0$. Then, $p(y) \leq -\gamma$, and we may derive an upper bound $-\gamma$ on $p(y)$.

In the context of Max-kSAT, a problem instance is expressed as degree-$2D$ polynomial optimization, which is then written as SOS to determine an upper bound. Similarly to the SDP approach, a rounding procedure may be used to identify a feasible solution and a lower bound. Rounding procedures typically involve random sampling similar to that in the GW algorithm. Details and an example are provided in Appendix A.

E. Approximation Guarantees and Caveats

To further contextualize the SDP and SOS approximation methods we can consider the simplest approximation algorithm for Max-kSAT: random guess, where each variable is assigned to be true with probability 1/2. This yields a $(1-(1/2)^k)$-approximation algorithm, that is, the approximated solution is expected to be at least $(1-(1/2)^k)$ times the optimum solution [9]. This is the baseline guarantee, and we hope to obtain solutions that are closer to the optimal with methods such as SDP and SOS.

While SOS provides a strict upper bound on the objective function, the rounded SOS solution has no approximation ratio guarantee. The approximation ratio given by the GW algorithm for Max-2SAT is lost in the generalization to Max-kSAT. Furthermore, the rounding procedure for SOS suffers from a lack of algebraic consistency. Algebraic consistency refers to the requirement that the element representing y_i and the element representing y_j multiply to equal the element representing y_iy_j in the polynomial basis b (Eq. (24)) [38].

For a polynomial objective function of degree-$2D$ and N variables, SOS requires the manipulation of matrices with dimension $N^{O(D)} \times N^{O(D)}$ due to the polynomial basis b. While an improvement over the brute force solution, this renders SOS intractable for large problem sizes, such as those found in industrial applications and scientific research.

III. THE PROPOSED ALGORITHM: HTAAC-QSOS

In this section we present our proposed algorithm: an SOS-inspired quantum metaheuristic for polynomial optimization with the Hadamard test and approximate amplitude constraints (HTAAC-QSOS). As a continuation of the previous discussion, we apply HTAAC-QSOS to Max-3SAT, simulating the algorithm and comparing it against classical benchmarks. We also show how HTAAC-QSOS for Max-3SAT generalizes to Max-kSAT. Finally, we describe a generalized overview of HTAAC-QSOS for any polynomial optimization problem.

A. HTAAC-QSOS for Max-3SAT

1. *Evaluating the Objective*

The polynomial objective function for Max-3SAT (Eq. (22)) may be rewritten as

$$\sum_{\ell} v(C_\ell) = \sum_{i>j>q>l} \Bigg(\left(W^{+,(1)}_{i,j} + W^{+,(2)}_{ij,ql}\right) - \left(y_i y_j W^{-,(1)}_{i,j} + y_i y_j y_q y_l W^{-,(2)}_{ij,ql}\right) \Bigg) \tag{25}$$

where

$$W^{+,(1)}_{i,j} = a^{(1)}_{i,j} + b^{(1)}_{i,j},\ W^{-,(1)}_{i,j} = a^{(1)}_{i,j} - b^{(1)}_{i,j},$$
$$W^{+,(2)}_{ij,ql} = a^{(2)}_{ij,ql} + b^{(2)}_{ij,ql},\ \text{and}\ W^{-,(2)}_{ij,ql} = a^{(2)}_{ij,ql} - b^{(2)}_{ij,ql}. \tag{26}$$

The variables $\{y_i\}_{i\in\{0,1,\ldots,N-1\}}$ are encoded using computational basis states, identically to HTAAC-QSDP. The objective function becomes

$$\frac{1}{2}\left(\overbrace{\left\langle W^{+,(1)}, \mathbf{1}_N\right\rangle + \left\langle W^{+,(2)}, \mathbf{1}_{N^2}\right\rangle}^{\text{constant terms}}\right) \tag{27}$$

$$-\frac{1}{2}\left(\underbrace{\left\langle W^{-,(1)}, \rho\right\rangle}_{\text{degree-2 terms}} + \underbrace{\left\langle W^{-,(2)}, \rho\otimes\rho\right\rangle}_{\text{degree-4 terms}}\right) \tag{28}$$

where $\mathbf{1}_N$ is an $N\times N$ matrix of all ones and $\mathbf{1}_{N^2}$ is an $N^2\times N^2$ matrix of all ones. The problem becomes

$$\begin{aligned} \underset{\rho}{\text{minimize}} \quad & \overbrace{\left\langle W^{-,(1)}, \rho\right\rangle}^{\text{degree-2 terms}} + \overbrace{\left\langle W^{-,(2)}, \rho\otimes\rho\right\rangle}^{\text{degree-4 terms}} \\ \text{subject to} \quad & \rho_{i,i} = 2^{-n}, \quad \forall i \le N \end{aligned} \tag{29}$$

where the objective function is the sum of two expectation values,

$$\begin{aligned} &\left\langle W^{-,(1)}, \rho\right\rangle + \left\langle W^{-,(2)}, \rho\otimes\rho\right\rangle \\ &= \langle\psi| W^{-,(1)} |\psi\rangle + \left(\langle\psi|\otimes\langle\psi|\right) W^{-,(2)} \left(|\psi\rangle\otimes|\psi\rangle\right). \end{aligned} \tag{30}$$

A core contribution of this work lies in the introduction of the product state for degree-4 terms. The product state $\rho\otimes\rho$ is a tensor product over two identical states ρ, such that its elements $(\rho\otimes\rho)_{ij,ql} = \rho_{i,j}\rho_{q,l}$ represent the degree-4 terms $y_i y_j y_q y_l$. This is the quantum analog to the polynomial basis in SOS. However, while SOS programming is solved with SDPs by including the polynomial basis as additional constraints, our proposed algorithm naturally preserves these constraints by encoding solutions as product states. HTAAC-QSOS therefore requires no additional constraints apart from those included in the original problem.

In terms of implementation, we propose a "multiple register" approach. Each "register" is an identical n-qubit quantum subsystem. Degree-2 terms are evaluated with a Hadamard test on a single register ρ and a unitary $U_{W^{-,(1)}} = e^{i\alpha W^{-,(1)}}$, similarly to Max-2SAT,

$$\langle\sigma^z_{\text{anc.}}\rangle_{W^{-,(1)}} = \text{Im}\left(\langle\psi| U_{W^{-,(1)}} |\psi\rangle\right). \tag{31}$$

To evaluate degree-4 terms in the objective function, two identical copies of the state ρ are prepared on two registers. A Hadamard test is implemented by operating on both registers with a larger $2n$-qubit unitary i.e., $U_{W^{-,(2)}} = e^{i\alpha W^{-,(2)}}$,

$$\langle\sigma^z_{\text{anc.}}\rangle_{W^{-,(2)}} \equiv \text{Im}\left(\left(\langle\psi|\otimes\langle\psi|\right) U_{W^{-,(2)}} \left(|\psi\rangle\otimes|\psi\rangle\right)\right). \tag{32}$$

The circuit diagram for these two Hadamard tests is shown in Figure 3, where $d=1$ and $d=2$ respectively.

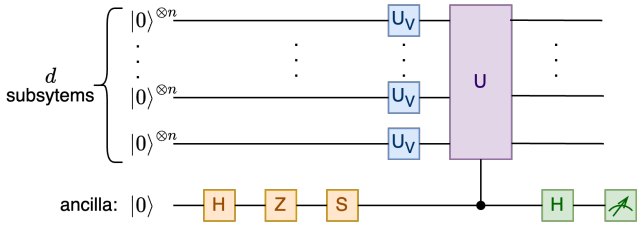

FIG. 3: **Hadamard test on nd qubits.** The value $\text{Im}\left(\langle\psi|^{\otimes d} U |\psi\rangle^{\otimes d}\right)$, where $|\psi\rangle \equiv U_V |0\rangle^{\otimes n}$, may be estimated by measuring a single ancilla qubit. Specifically, $\langle\sigma^z_{\text{anc.}}\rangle = \text{Im}\left(\langle\psi|^{\otimes d} U |\psi\rangle^{\otimes d}\right)$, where $\sigma^z_{\text{anc.}}$ is the Pauli-Z on the ancilla qubit. In HTAAC-QSDP we always take $d=1$. In HTAAC-QSOS, the values of d reflect the degree of the polynomial objective.

2. Approximate Amplitude Constraints

We want to enforce the constraint $\rho_{i,i} = 2^{-n}$ for all $i \leq N$. In Section II C 2, we describe how we may approximately enforce this constraint using $O(n^2)$ Pauli strings and a population-balancing unitary. Since we encode the objective function using product states, we do not require any additional constraints. Specifically, we write Pauli string equations

$$\begin{aligned} \langle \sigma_a^z, \rho \rangle &= 0, \forall a \leq n \\ \langle \sigma_a^z \sigma_b^z, \rho \rangle &= 0, \forall a < b \leq n \end{aligned} \tag{33}$$

and define the population-balancing unitary

$$\begin{aligned} U_P &= e^{i\beta P} \\ \text{where } P_{i,i} &= -\left(P_{\max} - \sum_j \left| W^{-,(1)}_{i,j} \right| \right), \\ P_{\max} &= \max_i \sum_j W^{-,(1)}_{i,j}. \end{aligned} \tag{34}$$

The population-balancing unitary has the same formulation as described in Section II C 2, but with $W^{-,(1)}$ in place of W^-. The purpose of the population-balancing unitary is to approximate the frequency at which a variable appears, and the larger matrix $W^{-,(2)}$ is therefore not required.

3. Retrieving Solutions

Extending the SEE method in Section II C 4, we evaluate the un-rounded solution using the quantum analog of Eq. (28)

$$\begin{aligned} &2^{n-1} \langle 0|^{\otimes n} H^{\otimes n} W^{+,(1)} H^{\otimes n} |0\rangle^{\otimes n} \\ &+ 2^{2n-1} \langle 0|^{\otimes 2n} H^{\otimes 2n} W^{+,(2)} H^{\otimes 2n} |0\rangle^{\otimes 2n} \\ &+ 2^{n-1} \langle \Psi^{(1)}| W^{-,(1)} |\Psi^{(1)}\rangle \\ &+ 2^{2n-1} \langle \Psi^{(2)}| W^{-,(2)} |\Psi^{(2)}\rangle . \end{aligned} \tag{35}$$

The first term is estimated using a Hadamard test with unitary $U_{W^{+,(1)}}$ acting on state $H^{\otimes n} |0\rangle^{\otimes n}$, and the second term is estimated using a Hadamard test $U_{W^{+,(2)}}$ acting on state $H^{\otimes 2n} |0\rangle^{\otimes 2n}$. The third and fourth terms are part of the objective function in Section III A 1. The un-rounded solution for Max-3SAT is

$$\begin{aligned} &\frac{2^{n-1}}{\alpha} \left(\langle \sigma^z_{\text{anc.}} \rangle^{\otimes H}_{W^{+,(1)}} - \langle \sigma^z_{\text{anc.}} \rangle_{W^{-,(1)}} \right) \\ &+ \frac{2^{2n-1}}{\alpha} \left(\langle \sigma^z_{\text{anc.}} \rangle^{\otimes H}_{W^{+,(2)}} - \langle \sigma^z_{\text{anc.}} \rangle_{W^{-,(2)}} \right). \end{aligned} \tag{36}$$

To obtain the rounded solution we require FST, and follow the same procedure outlined in Section II C 4.

B. Simulating HTAAC-QSOS for Max-3SAT

We test our proposed algorithm by simulating HTAAC-QSOS for Max-3SAT on a classical system and benchmarking results against classical SOS and heuristic methods. To present our results, we define observed performance, or the ratio between the HTAAC-QSOS result and the classical benchmark result for a given problem instance. In Tables I and II, this ratio is calculated and averaged over several instances of the same size.

1. SOS vs. HTAAC-QSOS

To test the performance of HTAAC-QSOS against SOS, we generate several Max-3SAT instances by drawing variables from a uniform distribution and removing clauses with repeating variables. The SOS techniques [31] are used to obtain classical solutions. We choose problems with 20 variables each because they may be approximated by SOS within reasonable time.

We remark that SOS often yields very tight upper bounds, frequently finding the optimal solution [9, 38]. However, since these upper bounds do not necessarily correspond to feasible solutions, we use a rounding procedure (described in Appendix A 2) to return the SDP solutions to the feasible region. We use the rounded result as a classical benchmark. The rounded result is typically within 92% to 96% of the upper bound for the problem sizes used [38]. Results are provided in Table I.

Both the SEE and rounded FST objective values from HTAAC-QSOS are consistently greater than that of the rounded classical SOS solution, despite encoding only a rough approximation of the problem. Notably, the un-rounded SEE objectives are less than their corresponding rounded FST objectives, implying favorable rounding behavior. We hypothesize that this rounding behavior can be attributed to the algebraic consistency maintained by HTAAC-QSOS. Algebraic consistency refers to the requirement that the element representing y_i and the element representing y_j multiply to equal the element representing $y_i y_j$ in the polynomial basis. This property is naturally preserved by the product states used in HTAAC-QSOS, but not by the rounding procedure in classical SOS. This results in a much simpler rounding scheme for HTAAC-QSOS, and one that yields heuristics that are closer to the optimal solution, based on our simulation result.

Moreover, in SDP relaxations, the rank of the solution may serve as an indicator of solution quality. Many polynomial optimization problems yield a natural SDP relaxation. If the SDP relaxation has a rank-1 solution, then the relaxation is exact, and there is no gap between the upper bound and the rounded solution. Intuitively, this suggests that rank-1 solutions are more desirable [36]. SDP relaxations often yield full-rank solutions. The solution to HTAAC-QSOS is always a pure state and therefore is rank-1, suggesting that HTAAC-QSOS may

Max-3SAT Problem Size		HTAAC-QSOS (SEE)		HTAAC-QSOS (FST)	
# of var.	# of clauses	Best Found	Average	Best Found	Average
20	80	1.040	1.039	1.084	1.078
	100	1.042	1.041	1.084	1.079
	120	1.025	1.024	1.071	1.067
	140	1.027	1.026	1.079	1.076
	160	1.004	1.003	1.050	1.045
	180	1.001	1.008	1.047	1.044

TABLE I: **Observed performance with respect to classical SOS**. Simulations of HTAAC-QSOS are used to approximate solutions for randomly-generated Max-3SAT instances with 20 variables and a given number of clauses. Both the sample-efficient estimation (SEE) and full state tomography (FST) methods for retrieving solutions are used. This is repeated over the space of loss hyperparameters, with the best and average solutions identified. For each Max-3SAT problem size, the four HTAAC-QSOS results (best SEE solution, average SEE solution, best FST solution, average FST solution) are divided by the rounded classical SOS solution and then averaged over multiple problem instances.

result in solutions that yield a smaller optimality gap when rounded.

2. *Classical Heuristic vs. HTAAC-QSOS*

To gauge the performance of HTAAC-QSOS for larger problems, we draw instances with 70 to 110 variables from the 2016 MaxSAT evaluation [26]. Since classical SOS is intractable for larger problems, the best known solutions are given by the winning heuristic solutions from the MaxSAT evaluation. These solvers primarily leverage local search algorithms and problem-specific heuristics. Like HTAAC-QSOS, these solvers prioritize speed over approximation ratio guarantees. The observed performance is therefore calculated with respect to these solutions, and averaged over instances of the same size given by the MaxSAT evaluation. The results are given in Table II, and Figure 4 shows loss and observed performance over 100 training epochs for a Max-3SAT instance with 110 variables and 1100 clauses.

The un-rounded SEE solutions' objective value are within ten percent of the objectives from the best classical heuristic, and the corresponding objectives of the rounded FST solutions are within three percent. Figure 4 shows that the un-rounded objectives, rounded objectives, and loss are strongly correlated. Un-rounded solutions consistently yield rounded FST solutions that are close to the optimal.

While these initial simulations of HTAAC-QSOS for Max-3SAT yield objectives that are slightly less than those given by existing classical heuristic solvers, the advantage of HTAAC-QSOS lies in its generalizability. Classical incomplete solvers may be specifically tailored to Max-SAT, but the HTAAC-QSOS framework can be generalized to any polynomial optimization problem. While HTAAC-QSOS works with even degree polynomials, this is without loss of generality because odd-degree polynomials can be made even through the introduction of an auxiliary variable such as y_0 [8, 9].

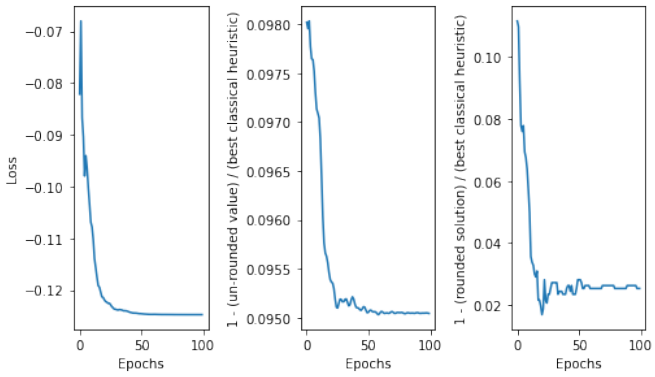

FIG. 4: **Loss and Observed Performance of HTAAC-QSOS over Epochs.** HTAAC-QSOS is used to find a heuristic solution for a Max-3SAT instance with 110 variables and 1100 clauses. These plots show loss (left), one minus the ratio between the un-rounded SEE solution and the best classical heuristic (middle), and one minus the ratio between the rounded FST solution and the best classical heuristic (right).

C. HTAAC-QSOS for Max-kSAT

The proposed SOS-inspired approach for Max-3SAT may be easily applied to Max-kSAT. The number of qubits required scales linearly with k, and the number of constraints is independent of k.

1. *Evaluating the Objective*

Max-kSAT can be expressed as a degree-$2D$ polynomial optimization problem, where $2D = 2\,\mathrm{ceil}(k/2)$ is the maximum degree of the polynomial objective[6]. The

[6] We remind the reader that Max-kSAT can also be expressed as a degree-k polynomial optimization problem, but we introduce

Max-3SAT Instance		HTAAC-QSOS (SEE)		HTAAC-QSOS (FST)	
# of var.	# of clauses	Best Found	Mean	Best Found	Mean
70	700	0.911	0.903	0.985	0.978
	800	0.909	0.901	0.984	0.981
	900	0.910	0.909	0.986	0.979
90	700	0.899	0.898	0.978	0.969
	800	0.903	0.902	0.979	0.972
	900	0.906	0.905	0.977	0.966
	1000	0.911	0.910	0.980	0.973
	1100	0.908	0.907	0.978	0.972
	1200	0.902	0.901	0.976	0.970
	1300	0.906	0.905	0.978	0.971
110	700	0.906	0.905	0.972	0.961
	800	0.909	0.908	0.974	0.961
	900	0.900	0.899	0.979	0.970
	1000	0.907	0.906	0.973	0.963
	1100	0.908	0.907	0.981	0.971

TABLE II: **Observed performance with respect to heuristic solvers**. Simulations of HTAAC-QSOS are used to approximate solutions for Max-3SAT instances with 70, 90, or 110 variables and a given number of clauses. Both the SEE and FST methods for retrieving solutions are used. This is repeated over the space of loss hyperparameters, with the best and average solutions identified. For each Max-3SAT problem size, the four HTAAC-QSOS results (best SEE solution, average SEE solution, best FST solution, average FST solution) are divided by the winning heuristic solution from the MaxSAT evaluation [26] and then averaged over multiple problem instances.

polynomial consists of only even-degree monomial terms of degree-$2d$, such that $d \in \{1, ..., D\}$. The generalized problem is written as

$$\begin{aligned} \underset{\rho}{\text{minimize}} \quad & \sum_{d=1}^{D} \left\langle W^{-,(d)}, \rho^{\otimes d} \right\rangle \\ \text{subject to} \quad & \rho_{i,i} = 2^{-n}, \quad \forall i \leq N \end{aligned} \tag{37}$$

where the objective function is a sum of expectation values

$$\sum_{d=1}^{D} \left\langle W^{-,(d)}, \rho^{\otimes d} \right\rangle = \sum_{d=1}^{D} \langle \Psi^{(d)} | W^{-,(d)} | \Psi^{(d)} \rangle \tag{38}$$

and we define $|\Psi^{(d)}\rangle \equiv |\psi\rangle^{\otimes d}$ to simplify notation.

Each $\langle \Psi^{(d)} | W^{-,(d)} | \Psi^{(d)} \rangle$ encodes terms of degree-$2d$ and can be estimated using the Hadamard test,

$$\alpha \langle \Psi^{(d)} | W^{-,(d)} | \Psi^{(d)} \rangle \approx \mathrm{Im}(\langle \Psi^{(d)} | U_{W^{-,(d)}} | \Psi^{(d)} \rangle) \tag{39}$$

where $U_{W^{-,(d)}} \equiv e^{i\alpha W^{-,(d)}}$ (see Figure 3). To be specific, we evaluate all terms of degree-$2d$ by first simultaneously and separately preparing d identical copies of ρ. Then, we use the nd-qubit unitary $U_{W^{-,(d)}}$ to conduct the Hadamard test upon the entire quantum system – all d copies. Repeating this for all possible values of $d \in \{1, ..., D\}$, we can evaluate all terms in the objective function. We therefore require a total of $O(D) = O(k)$ Hadamard tests and $O(nD) = O(nk)$ qubits to estimate the objective function.

This is our "multiple register" approach, where each register of n qubits hosts a copy of ρ. This allows us to generalize the original HTAAC-QSDP framework for degree-k polynomial optimization with $O(k)$ time and resource complexity.

2. *Approximate Amplitude Constraints*

The Pauli strings and the population-balancing unitary are formulated identically to those in Max-3SAT (see Section III A 2), since the equality constraint in the optimization problem is the same. Thus, the number of measurements required to approximately enforce constraints is independent of k.

3. *Retrieving Solutions*

The un-rounded SEE solution for Max-3SAT in Eq. (36) can be generalized to

$$\sum_{d=1}^{D} \frac{2^{nd-1}}{\alpha} \left(\langle \sigma^z_{\text{anc.}} \rangle^{\otimes H}_{W^{+,(d)}} - \langle \sigma^z_{\text{anc.}} \rangle_{W^{-,(d)}} \right), \tag{40}$$

where we recall $D = \text{ceil}(k/2)$ and each term may be evaluated with a Hadamard test. We therefore require $O(k) = O(D)$ Hadamard tests to determine this un-rounded solution.

The second rounding method follows the same procedure described in Section II C 4. This method is the same

y_0 here to make the polynomial even and therefore simpler to address with our quantum approach.

for all values of k, and involves full state tomography the n-qubit subsystem.

D. HTAAC-QSOS for Polynomial Optimization

We recall that any degree-k polynomial optimization can be made into a degree-$2D$ polynomial with terms of degree $2d$, where $d \in \{1, ..., D\}$, using a auxiliary variable [8, 9]. For the quadratic optimization problems addressed with HTAAC-QSDP, $D = 1$ and $d = 1$. In the polynomial optimization problems we address with our proposed HTAAC-QSOS, we allow $D \geq 1$ and $d \in \{1, ..., D\}$. Terms of degree-$2d$ may be approximated with d quantum subsystems of n qubits each. Therefore, we require $O(nD) = O(nk)$ qubits to estimate the objective function. Furthermore, the property of Pauli strings to form a complete basis suggests that they may be used to enforce any linear equality constraint. To retrieve the solutions, the SEE and FST methods may be generalized. The SEE method requires $O(D) = O(k)$ Hadamard tests, namely, one for value of $d \in \{1, ..., D\}$. The FST rounding method for polynomial optimization over binary variables requires full state tomography of a n-qubit subsystem, but is constant in k.

IV. CONCLUSIONS AND OUTLOOK

In this manuscript, we propose a SOS-inspired quantum metaheuristic for polynomial optimization (HTAAC-QSOS), that extends the recently proposed HTAAC-QSDP algorithm [1] using classical sum-of-squares (SOS). This technique yields a heuristic that is scalable to challenging and useful problems, such as those found in operations research and similar research applications. Moreover, HTAAC-QSOS is generalizable, and scales linearly with the degree of the polynomial. Notably, HTAAC-QSOS exhibits superior rounding behavior compared to classical SOS methods, likely due to its preservation of algebraic consistency through product state encoding.

In our study of Max-kSAT, we emphasize as the degree k of the polynomial objective increases, only the evaluation of the objective changes. In fact, the number of approximate amplitude constraints required in Max-kSAT is independent of the degree of the polynomial objective. While classical SOS requires additional constraints with the increase of k, these constraints are naturally conserved by the product state encoding in HTAAC-QSOS. If the Hadamard tests are conducted sequentially on a set of $D = \text{ceil}(k/2)$ registers, then the time and resource complexities of HTAAC-QSOS are therefore only linear in k, whereas the less-favorable complexities of classical SOS are polynomial. Alternatively, if we want Hadamard tests to be conducted simultaneously on more registers, then HTAAC-QSOS is constant in time and quadratic in resources (i.e., qubits) with respect to k. HTAAC-QSOS thus has the potential to address a broader class of polynomial optimization problems than previous quantum approaches or problem-specific classical heuristics, while maintaining better scaling than classical SOS methods as the degree of the polynomial increases.

The generalization to any polynomial optimization may be tested by applying the HTAAC-QSOS approach to other families of polynomial optimization problems, including those that are optimized over *continuous* variables. In these continous cases, the rounding step of the FST solution method may not be required. One possible application is in quantum chemistry, where the SOS hierarchy (i.e., the reduced density matrix method) is used to study weak coupling perturbation theory [39]. SOS is also used as an approach towards the best separable quantum state problem [40].

Additionally, the high performance of the FST rounding method in classical simulations suggests that HTAAC-QSOS may be a strong candidate for a quantum-inspired algorithm. Since the circuit does not need to be implemented on qubits, the Hadamard test is not required, and we may directly evaluate the objective function using the weight matrix.

Acknowledgements

R.B. was supported by NSF CCF (grant #1918549). S.F.Y. would like to thank the NSF via QIdeas HDR (OAC-2118310) and the CUA PFC (PHY-2317134).

[1] T. L. Patti, J. Kossaifi, A. Anandkumar, and S. F. Yelin, Quantum goemans-williamson algorithm with the hadamard test and approximate amplitude constraints, Quantum **7**, 1057 (2023).

[2] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195–202 (2017).

[3] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, Quantum chemistry in the age of quantum computing, Chemical Reviews **119**, 10856–10915 (2019).

[4] A. Montanaro, Quantum algorithms: an overview, npj Quantum Information **2**, 1–8 (2016).

[5] J. Preskill, Quantum computing in the nisq era and beyond, Quantum **2**, 79 (2018).

[6] B. Korte and J. Vygen, Introduction, in *Combinatorial Optimization: Theory and Algorithms* (Springer, Berlin, Heidelberg, 2018) p. 1–13.

[7] J. Berg, A. Hyttinen, and M. Järvisalo, Applications of maxsat in data analysis, in *EPiC Series in Computing*, Vol. 59 (EasyChair, 2019) p. 50–64.

[8] L. Vandenberghe and S. Boyd, Semidefinite programming, SIAM Review **38**, 49–95 (1996).

[9] V. V. Vazirani, *Approximation Algorithms* (Springer Link, 2001).

[10] F. G. S. L. Brandão, R. Kueng, and D. S. França, Faster quantum and classical sdp approximations for quadratic binary optimization, Quantum **6**, 625 (2022).

[11] F. G. Brandao and K. M. Svore, Quantum speed-ups for solving semidefinite programs, in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (2017) p. 415–426.

[12] J. v. Apeldoorn, A. Gilyén, S. Gribling, and R. d. Wolf, Quantum sdp-solvers: Better upper and lower bounds, Quantum **4**, 230 (2020).

[13] J. van Apeldoorn and A. Gilyén, Improvements in quantum sdp-solving with applications, in *DROPS-IDN/v2/document/10.4230/LIPIcs.ICALP.2019.99* (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019).

[14] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin, Quantum optimization of maximum independent set using rydberg atom arrays, Science **376**, 1209–1215 (2022).

[15] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, Quantum sdp solvers: Large speed-ups, optimality, and applications to quantum learning, in *DROPS-IDN/v2/document/10.4230/LIPIcs.ICALP.2019.27* (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019).

[16] D. Patel, P. J. Coles, and M. M. Wilde, Variational quantum algorithms for semidefinite programming, arXiv 10.48550/arXiv.2112.08859 (2021), arXiv:2112.08859 [quant-ph].

[17] T. Albash and D. A. Lidar, Adiabatic quantum computation, Reviews of Modern Physics **90**, 015002 (2018).

[18] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv 10.48550/arXiv.1411.4028 (2014).

[19] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, arXiv 10.48550/arXiv.quant-ph/0001106 (2000).

[20] E. Gibney, D-wave upgrade: How scientists are using the world's most controversial quantum computer, Nature **541**, 447–448 (2017).

[21] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse ising model, Physical Review E **58**, 5355–5363 (1998).

[22] J. M. Arrazola, V. Bergholm, K. Brádler, T. R. Bromley, M. J. Collins, I. Dhand, A. Fumagalli, T. Gerrits, A. Goussev, L. G. Helt, J. Hundal, T. Isacsson, R. B. Israel, J. Izaac, S. Jahangiri, R. Janik, N. Killoran, S. P. Kumar, J. Lavoie, A. E. Lita, D. H. Mahler, M. Menotti, B. Morrison, S. W. Nam, L. Neuhaus, H. Y. Qi, N. Quesada, A. Repingon, K. K. Sabapathy, M. Schuld, D. Su, J. Swinarton, A. Száva, K. Tan, P. Tan, V. D. Vaidya, Z. Vernon, Z. Zabaneh, and Y. Zhang, Quantum circuits with many photons on a programmable nanophotonic chip, Nature **591**, 54–60 (2021).

[23] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nature Reviews Physics **3**, 625–644 (2021).

[24] L. Bittel and M. Kliesch, Training variational quantum algorithms is np-hard, Physical Review Letters **127**, 120502 (2021).

[25] S. Cai and X. Zhang, Pure maxsat and its applications to combinatorial optimization via linear local search, in *Principles and Practice of Constraint Programming*, edited by H. Simonis (Springer International Publishing, Cham, 2020) p. 90–106.

[26] O. of the 19th International Conference on Theory and A. of Satisfiability Testing, The homepage of eighth max-sat evaluation, `http://maxsat.ia.udl.cat/introduction/` (2016).

[27] P. A. Parrilo, Semidefinite programming relaxations for semialgebraic problems, Mathematical Programming **96**, 293–320 (2003).

[28] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, Journal of the ACM **42**, 1115–1145 (1995).

[29] R. Hickey and F. Bacchus, Large neighbourhood search for anytime maxsat solving, in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, edited by L. D. Raedt (ijcai.org, 2022) pp. 1818–1824.

[30] A. A. Ahmadi, Sum of squares (sos) techniques: An introduction.

[31] S. Prajna, A. Papachristodoulou, and P. Parrilo, Introducing sostools: a general purpose sum of squares programming solver, in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, Vol. 1 (2002) p. 741–746 vol.1.

[32] B. Harrach, Solving an inverse elliptic coefficient problem by convex non-linear semidefinite programming, Optimization Letters **16**, 1599–1609 (2022).

[33] A. Gepp, G. Harris, and B. Vanstone, Financial applications of semidefinite programming: a review and call for interdisciplinary research, Accounting and Finance **60**, 3527–3555 (2020).

[34] S. A. Cook, The complexity of theorem-proving procedures, in *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71 (Association for Computing Machinery, New York, NY, USA, 1971) p. 151–158.

[35] C. Howson, *Logic with Trees: An Introduction to Symbolic Logic* (Routledge, New York, 1997).

[36] A. Lemon, A. M.-C. So, and Y. Ye, Low-rank semidefinite programming: Theory and applications, Foundations and Trends in Optimization **2**, 1–156 (2016).

[37] S. Hakemi, M. Houshmand, E. KheirKhah, and S. A. Hosseini, A review of recent advances in quantum-inspired metaheuristics, Evolutionary Intelligence **17**, 627–642 (2024).

[38] H. van Maaren, L. van Norden, and M. J. H. Heule, Sums of squares based approximation algorithms for max-sat, Discrete Applied Mathematics **156**, 1754–1779 (2008).

[39] M. B. Hastings, Perturbation theory and the sum of squares, arXiv (2024), arXiv:2205.12325 [cond-mat, physics:hep-th, physics:quant-ph].

[40] A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri, Complete family of separability criteria, Physical Review A **69**, 022308 (2004).

Appendix A: Sum-of-Squares (SOS) and Rounding Procedures

In this section of the appendix, we provide more details on the sum-of-squares (SOS) method as well as its rounding procedures.

1. SOS Hierarchy

While there exist several relaxation algorithms and rounding procedures used to approximate solutions to Max-kSAT and similar problems, we choose SOS as the most standard and systematic method as a classical benchmark for our algorithm.

Let $F(y)$ be a multivariate polynomial, where y denotes the N variables $\{y_i\}_{i\in\{0,1,\ldots,N-1\}}$. The goal of classical SOS is to write $F(y)$ in such a way that the nonnegativity of $F(y)$ becomes obvious. Specifically, the existence of an SOS decomposition

$$F(y) = \sum_j q_j^2(y), \tag{A1}$$

where $q_j(y)$ are some polynomials in y, is an algebraic certificate of nonnegativity.

The SOS hierarchy is a hierarchy is convex relaxations, ordered with increasing power and computational cost. The Dth level of the hierarchy corresponds to polynomial optimization problems with a degree-$2D$ polynomial objective. The first level of the hierarchy, where $D = 1$, corresponds to an SDP for a quadratic optimization problem. To consider higher-degree polynomials, new variables and constraints must be introduced such that the problem may then be solved as a larger SDP with additional constraints. Specifically, we draw a connection between SOS and semidefinite matrices with the following theorem.

Theorem 1. *A degree-$2D$ multivariate polynomial $F(y)$ is a sum-of-squares if and only if there exists positive semidefinite matrix Q such that*

$$F(y) = b^T Q b \tag{A2}$$

where $b = \begin{pmatrix}1 & y_1 & y_2 & \ldots & y_N & y_1y_2 & \ldots & y_N^D\end{pmatrix}^T$. The vector b is the polynomial basis, and the matrix Q is called the Gram matrix.

A proof is provided by [30].

To show how SOS is applied to polynomial optimization problems, we consider a generic problem of the form

$$\begin{aligned} &\text{maximize} && p(y) \\ &\text{subject to} && h_j(y) = 0, \quad \forall j \in \{1, 2, \ldots, M\} \end{aligned} \tag{A3}$$

where $p(y)$ is a polynomial objective that is not necessarily convex, $h_j(y) = 0$ are equality constraints, and M is the number of constraints. Using the given problem, we define a new polynomial[7]

$$F(y) \equiv -p(y) - \gamma + \sum_j t_j(y) h_j(y), \tag{A4}$$

where γ a real-valued variable, and $t_j(y)$ is a polynomial with its degree limited by D. If $F(y)$ may be factored into a sum of squares for a some value of γ and some polynomials $t_j(y)$, we guarantee that $F(y) \geq 0$. When the constraints are satisfied such that $h_j(y) = 0$, we obtain $p(y) \leq -\gamma$ such that $-\gamma$ forms an upper bound on $p(y)$. Formally, we want to obtain a tight upper bound bound on $p(y)$ by solving

$$\begin{aligned} &\text{maximize} && \gamma \\ &\text{subject to} && F(y) \text{ is SOS.} \end{aligned} \tag{A5}$$

Since we look for solutions where $h_j(y) = 0$, the polynomials $t_j(y)$ in Eq. (A4) function as additional degrees of freedom that allow tighter upper bounds.

As a very simple example, let us consider a Max-3SAT problem with two clauses: $(x_1 \vee x_2 \vee x_3)$ and $(\neg x_1 \vee \neg x_2 \vee x_3)$. The equivalent polynomial optimization problem is

$$\begin{aligned} &\text{maximize} && \frac{1}{4}\left(y_0y_3 - y_1y_2 + y_0y_1y_2y_3\right) + \frac{7}{4} \\ &\text{subject to} && y_i^2 = 1 \, \forall i \in \{0, 1, 2, 3\} \end{aligned} \tag{A6}$$

where the objective function is formulated using Eq. (21). The result may be approximated using the SOS in Eq. (A5), where

$$\begin{aligned} F(y) \equiv &-\frac{1}{4}\left(y_0y_3 - y_1y_2 + y_0y_1y_2y_3\right) - \frac{7}{4} \\ &- \gamma + \sum_{i=0}^{3} t_i(y)(y_i^2 - 1). \end{aligned} \tag{A7}$$

This is results in a polynomially-sized SDP by Theorem 1. Specifically, we define

$$b \equiv \begin{pmatrix}1 & y_0 & y_1 & y_2 & y_3 & y_0y_1 & y_2y_3\end{pmatrix}^T \tag{A8}$$

and determine Q as a function of γ and the coefficients of $t_i(y)$ such that $F(y)$ is SOS, that is, $F(y) = b^T Q b = \langle bb^T, Q^T\rangle$. In this way, the problem becomes the SDP

$$\begin{aligned} &\underset{Q \in \mathbb{S}^+}{\text{maximize}} && \langle W, Q\rangle \\ &\text{subject to} && \langle bb^T, Q\rangle = F(y) \end{aligned} \tag{A9}$$

where W is defined such that $\langle W, Q\rangle \propto \gamma$. Solving the SDP, we obtain an upper bound $-\gamma = 2$, as we expect.

[7] SOS is often used to determine a lower bound on the solution of the original problem. Here, we want to find an upper bound. To make the notation more straightforward, we introduced a factor of -1 to $p(y)$ in our formulation of $F(y)$.

2. Rounding Procedures

The rounding procedure used in our SOS solution is equivalent one of the rounding procedures described by van Maaren et al. [38], and the steps are as follows. It can be shown that the optimal solution Q of the SOS method described in the previous section has an eigenvalue zero. We determine the orthogonal basis of eigenvectors $v_1, ..., v_L$ of Q that have eigenvalue zero, such that $v_l^T Q v_l = 0$ for $l \in \{1, ..., L\}$. Then, we randomly sample a vector $\lambda = \begin{pmatrix} \lambda_1 & ... & \lambda_L \end{pmatrix}$ from the L-dimensional unit sphere and use this to generate a linear combination of these eigenvectors,

$$P = \sum_{l=1}^{l} \lambda_l v_l. \tag{A10}$$

The elements of P are rounded to $+1$ or -1, and the results are used as the solution $\{y_i\}_{i \in \{0,1,...,N-1\}}$ to the original polynomial optimization problem.

We note that the vectors v_l for $l \in \{1, ..., L\}$ do not preserve algebraic consistency, that is, the product of elements corresponding to y_i and y_j do not necessarily equal the element corresponding to $y_i y_j$. There are methods to mitigate this, such as those developed by Ref. [38]. In their experiments using Max-3SAT problems with 20 variables, the rounded result reaches 97.2% of the upper bound.