Will a Quantum-Inspired Classical Algorithm Become the Default Algorithm for an Industrially Relevant Computational Task in the Next Decade?

Adam Godel[1] and Egemen Tunca[1]

[1] *Department of Physics, Boston University, Boston, Massachusetts 02215, USA*

(Dated: December 14, 2025)

We examine the viability of two quantum-inspired classical algorithms—one for linear algebra and recommendation systems and the other for Grover search—for solving computational problems more efficiently than the current optimal classical algorithms. The argument in favor primarily emphasizes the fact that these algorithms can apply to a very wide set of problems relatively efficiently. Experimental results indicate that quantum-inspired algorithms can be highly accurate and efficient in solving low-rank linear systems of equations. Certain optimal search algorithms where Grover's algorithm functions as a subroutine may be promising applications for quantum-inspired Grover's algorithm. The argument against underscores the costly polynomial scaling of the quantum-inspired algorithms for linear algebra and recommendation systems and the general tensor network construction inherent to quantum-inspired Grover's algorithm unlikely to be more optimal than a specific construction per problem. When we presented our research to our quantum computing class, their opinion was split but more students ultimately voted against the proposition than in favor.

I. INTRODUCTION

Quantum-inspired classical algorithms have served as a valuable tool for benchmarking the performance of quantum computers in reality. These algorithms are generally a variation of a quantum algorithm particularly suitable for execution on a classical computer.

However, some of these algorithms present novel and potentially competitive approaches compared to other classical algorithms, separate from the quantum algorithm by which they were inspired. We present our investigation of the viability of these algorithms in their own right to argue if one of them could become the primary algorithm for some industrial use case.

This proposition is worthwhile to explore because, if it is true, it gives us a reason to study quantum computation completely independently of the viability of practically realizable and scalable quantum hardware. Since large-scale quantum hardware is not yet viable, but the field of research relating to quantum-inspired classical algorithms is active and growing, it is an interesting time to examine this proposition.

We will establish our argument through the discussion of two algorithms: one that can solve linear systems of equations and compute preferences in recommendation systems [1], and one that can perform a search of length n bitstrings to find one or more marked states [2].

II. TECHNICAL BACKGROUND

A. Quantum-inspired singular value decomposition

We begin by discussing quantum-inspired algorithms for linear systems of equations and recommendation systems, as described in Ref. [1].

Given an $m \times n$ matrix $A \in \mathbb{R}^{m\times n}$ with singular value decomposition (SVD)

$$A = \sum_{\ell=1}^{k} \sigma_\ell \boldsymbol{u}^{(\ell)} \boldsymbol{v}^{(\ell)^T}, \tag{1}$$

the goal is to sample entries of the n-dimensional vector

$$\boldsymbol{x} = \sum_{\ell=1}^{k} \lambda_\ell \boldsymbol{v}^{(\ell)}, \tag{2}$$

with respect to the length-square probability distribution $p_x(i) = x_i^2/\|\boldsymbol{x}\|^2$. For linear systems of equations, we set

$$\lambda_\ell = \frac{1}{\sigma_\ell^2} \langle \boldsymbol{v}^{(\ell)}, A^T \boldsymbol{b} \rangle, \tag{3}$$

so that $A\boldsymbol{x} = \boldsymbol{b}$. For recommendation systems, we consider A to be the preference matrix whose entries A_{ij} denote the preference (or alignment) of item i to item j and set

$$\lambda_\ell = \langle A_i^T, \boldsymbol{v}^{(\ell)} \rangle, \tag{4}$$

where A_i is the i-th row of A.

We first compute the approximate singular values $\tilde{\sigma}_\ell$ and approximate right singular vectors $\tilde{\boldsymbol{v}}^{(\ell)}$ of A using the Frieze-Kannan-Vempala (FKV) algorithm [3], as outlined in Figure 1. This algorithm works by constructing a smaller matrix $C \in \mathbb{R}^{r\times c}$ by sampling r rows and c columns of A, where r and c are smaller but otherwise independent of m and n, respectively.

Specifically, we compute the Frobenius norm of A, defined as

$$\|A\|_F \equiv \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |A_{ij}|^2}, \tag{5}$$

as well as the norm of A_i to construct length-square probability distributions $p(i) = \|A_i\|^2/\|A\|_F^2$ and $q_i(j) =$

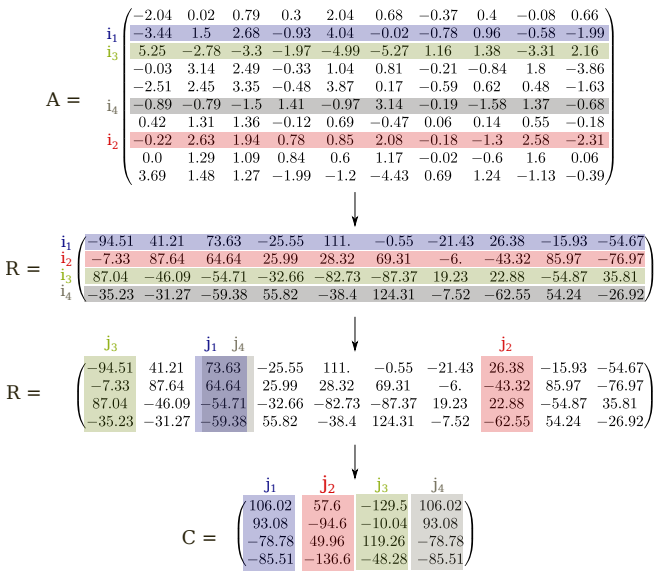

FIG. 1. Schematic representation of the FKV algorithm, as constructed in Ref. [1]. The singular values of C approximate those of A, while the singular vectors of C can be used to reconstruct approximations to the singular vectors of A.

$|A_{ij}|^2/\|A_i\|^2$. We sample r row indices from the row distribution $p(i)$, and then select an index $s \in \{1, 2, \ldots, r\}$ uniformly at random to sample a column index j from the column distribution $q_{i_s}(j)$. We renormalize the rows and columns respectively after sampling them.

The characteristic that makes this algorithm "quantum-inspired" is the way we store our data: we do not directly compute the vectors, but rather query their entries in accordance with a length-square distribution, a process reminiscent of quantum measurement.

Specifically, consider the fact that the coefficients λ in $\boldsymbol{x}$ will always be inner products, so we can write them as $\lambda = \langle \boldsymbol{y}, \boldsymbol{z} \rangle = \sum_{i=1}^{n} y_i z_i$ for appropriate vectors $\boldsymbol{y}, \boldsymbol{z}$ as defined for the specific application. We then define the random variable χ that takes values $\chi_i = y_i z_i / p_y(i)$, where the indices i are sampled from the length-square distribution of one of the vectors (say, $\boldsymbol{y}$) giving $p_y(i) = y_i^2/\|\boldsymbol{y}\|^2$. The expected value of this random variable is

$$\mathbb{E}(\chi) = \sum_{i=1}^{n} \frac{y_i z_i}{p_y(i)} p_y(i) = \langle \boldsymbol{y}, \boldsymbol{z} \rangle = \lambda, \tag{6}$$

so we can draw N samples $\chi^{(1)}, \chi^{(2)}, \ldots, \chi^{(N)}$ and compute the unbiased estimator $\hat{\lambda} = \frac{1}{N} \sum_{j=1}^{N} \chi^{(j)} \approx \lambda$.

We now have approximate singular values $\tilde{\sigma}_\ell$ and approximate coefficients $\tilde{\lambda}_\ell$. Using the left singular vectors $\boldsymbol{\omega}^{(\ell)}$ of C, we can implicitly calculate the approximate solution vector $\tilde{\boldsymbol{x}} = \sum_{\ell=1}^{k} \lambda_\ell \boldsymbol{v}^{(\ell)} = R^T \boldsymbol{w}$, where

$$\boldsymbol{w} \equiv \sum_{\ell=1}^{k} \frac{\tilde{\lambda}_\ell}{\tilde{\sigma}_\ell} \boldsymbol{\omega}^{(\ell)}. \tag{7}$$

We perform rejection sampling [4] to sample from these vectors using only query access to the entries of $\boldsymbol{w}$ and R. The idea is that given sampling access to some distribution $\mathcal{P}$, rejection sampling allows us to sample from a "close" distribution $\mathcal{Q}$ by pulling a sample s from $\mathcal{P}$, computing $r_s = \frac{\mathcal{Q}(s)}{m\mathcal{P}(s)}$ for some constant m, and then outputting s with probability r_s or restarting otherwise.

We choose $\mathcal{P}$ to be the distribution formed by first sampling a row index i uniformly at random and then sampling a column index j from the length-square distribution $q_i(j) = |R_{ij}|^2/\|R_i\|^2$. $\mathcal{Q}$ is the distribution formed from sampling $\boldsymbol{w}$. This means that we should output j with probability

$$r_j = \frac{|\langle \boldsymbol{w}, R_{\cdot,j} \rangle|^2}{\|R_{\cdot,j}\|^2 \|\boldsymbol{w}\|^2}. \tag{8}$$

Overall, this gives us a technique to sample entries of an approximate vector formed as a linear combination of the right singular vectors of some matrix with specific coefficients λ. This allows us to classically solve systems of linear equations and compute preferences in recommendation systems, but in a "quantum-inspired" way due to the length-square distribution and query access restrictions that make the algorithm efficient.

B. Quantum-inspired Grover's algorithm

We continue with a discussion of quantum-inspired Grover's algorithm (QiGA), as constructed in Ref. [2]. Let $\{w^1, w^2, \ldots, w^S\} \in \{0,1\}^n$ be the set of "marked" bitstrings of length n, of which we are searching for one. Let $|w\rangle = \sum_{\alpha=1}^{S} |w^\alpha\rangle$ be the uniform superposition of marked states. Recall that the Grover oracle is

$$U_w = 1 - 2|w\rangle\langle w|. \tag{9}$$

Let $|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$ be the equal superposition state. We want to classically compute the state $|\Psi_w\rangle = U_w|s\rangle$. We can express $|\Psi_w\rangle$ as a matrix product state (MPS), a type of tensor network, with a relatively small bond dimension χ. An MPS is expressed as

$$|\Psi\rangle = \sum_{x_{n-1}\cdots x_0 \in \{0,1\}^n} M_{n-1}(x_{n-1}) \cdots M_0(x_0) |x_{n-1} \cdots x_0\rangle, \tag{10}$$

where the $M_i(x)$ are $\chi \times \chi$ matrices for $1 \le i \le n-2$; $M_0(x)$ is a $\chi \times 1$ matrix and $M_{n-1}(x)$ is a $1 \times \chi$ matrix.

The key observation is that the cost of constructing $|\Psi_w\rangle$ as an MPS depends on the entanglement generated by the particular oracle U_w, and the entanglement in the quantum state in between the two oracle calls generated by Grover's algorithm is usually very low. An MPS can be sampled in any local basis using a recursive algorithm with runtime $O(n\chi^2)$ [5], with χ as the bond dimension, which scales with the entanglement of the state.

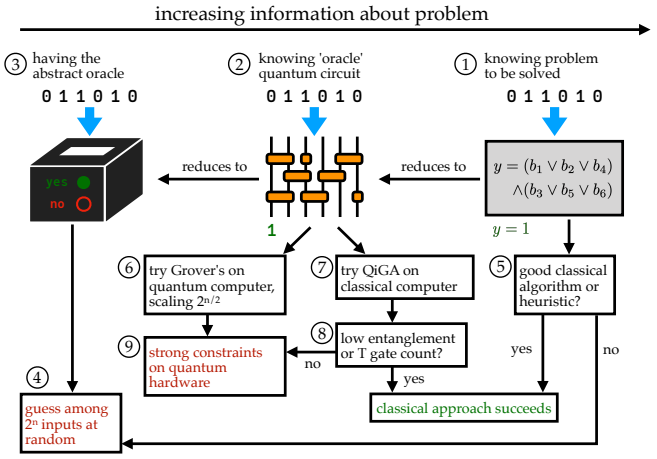

FIG. 2. Diagram indicating when it is suitable to use QiGA, as per Ref. [2]. Problems where we know the oracle quantum circuit, and it has low entanglement and can be simulated easily, are most suitable for the algorithm.

The MPS construction turns out to be the bottleneck of QiGA. We can express the state as

$$|\Psi_w\rangle = |s\rangle - \frac{2}{\sqrt{2^n}} \sum_{\alpha=1}^{S} |w^\alpha\rangle, \tag{11}$$

which we know is an MPS of bond dimension $\chi = 1 + S$ since it's the sum of $1 + S$ product states. We can then perform a *non-unitary classical operation*, subtracting $|s\rangle$ from the state above.

We can prove that a mapping from $|\Psi\rangle \otimes |s\rangle$ to $(|\Psi\rangle - |s\rangle) \otimes |s\rangle$ for all $|\Psi\rangle$ is non-unitary in a similar way we can prove the no-cloning theorem: in the case where $|\Psi\rangle = |s\rangle$, this will clearly not preserve norm. Applying this operation to an MPS costs $O(n\chi^3)$ [2].

Once we normalize this new state $|W\rangle = \frac{|\Psi_w\rangle - |s\rangle}{\||\Psi_w\rangle - |s\rangle\|}$, we can sample from it to obtain the states $\{|w^\alpha\rangle\}$ with uniform probability. In fact, we can even use a similar approach to count the number of solutions by acting with U_w on the unnormalized state $\sum_{b=0}^{2^n-1} |b\rangle$. Then the squared norm of the resulting state after we subtract $|s\rangle$ gives the number of solutions.

While Ref. [2] focuses primarily on the 3-SAT algorithm as an example, an important thing to note about QiGA is that it is as general as Grover's algorithm. Consider Figure 2, which illustrates that QiGA can be used in essentially any case where Grover's algorithm can be used. As long as we can construct the quantum oracle for any Grover problem, we can run QiGA.

III. THE CASE FOR

The power of these two quantum-inspired algorithms comes from their *generality*. These algorithms can be applied to a wider range of problems than direct computation or most classical heuristics, often with comparable performance. While the existence of scalable and widespread quantum hardware in a decade would not invalidate our argument, we find the premise unlikely. Furthermore, the practical superiority of the comparable quantum algorithms—the Harrow-Hassidim-Lloyd (HHL) algorithm for solving linear systems and Grover's algorithm for search—is questionable at best [2, 6].

A. Examining quantum-inspired SVD in practice

Ref. [1] provides several numerical experiments of their proposed quantum-inspired algorithms for linear algebra and recommendation systems. The setup that provided the greatest speedup over direct SVD calculation was applying the algorithm to solve linear systems of equations $A\boldsymbol{x} = \boldsymbol{b}$ for randomly chosen $A \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, where $m = 40{,}000$ and $n = 20{,}000$.

The results of this experiment are indicated in Figure 3. For $k = \kappa = 5$, it took 2470.4 seconds for the quantum-inspired algorithm to execute compared to 5192.5 seconds for a direct calculation on two Intel Xeon CPUs operating at 2.4GHz with access to 252GB of shared memory in Python. The overall error in $\boldsymbol{x}$ compared to the exact solution was 8.7%. These results are promising, as they indicate that with high enough dimension and low enough rank and condition number, a speedup will be present.

While in the experiment for recommendation systems, which used the MovieLens 100K database, the quantum-inspired algorithm was much slower than direct calculation, the dataset used was relatively small, containing only 611 users and 9724 movies [1].

The algorithm is better suited for much larger datasets that could be represented with very low-rank approximations. Ref. [1] cites running the algorithm on powerful supercomputers to compute much larger datasets as a future goal to more fully evaluate the algorithm's practical efficiency.

Useful recommendation systems like this do exist in the real world. For example, consider the latent semantic indexing (LSI) approach for classifying, updating, and retrieving patent documents described in Ref. [7]. This problem relies on SVD of a low-rank approximation of some preference matrix representing some aspect of data relating to filed patents. For example, a *terms-documents* matrix A contains entries A_{ij} that represent the number of times term i appears in document j.

Each class of patents contains thousands of documents and tens of thousands of words, overall creating a much larger matrix than the one representing the MovieLens 100K database, and Ref. [7] indicates that they can be approximated with a rank as low as $k = 80$. This may indicate that the quantum-inspired algorithm for recommendation systems may provide a speedup in comparison to directly calculating SVD for LSI.

The argument for the utility of quantum-inspired algo-

Parameters				Error				
Case study	r	c	N	η_σ	η_A	η_{A^+}	η_λ	$\eta_{\boldsymbol{x}}$
Random matrix	4250	4250	10^4	0.010 ± 0.005	0.028 ± 0.004	0.101 ± 0.027	0.387 ± 0.191	0.087 ± 0.053

FIG. 3. Relative errors with corresponding standard deviations, as simulated in Ref. [1], for $m = 40{,}000, n = 20{,}000, k = \kappa = 5$. Each coefficient λ_ℓ was estimated by taking $N = 10^4$ samples and producing condensed matrices of size $r = 4250$ and $c = 4250$. The experiment was repeated 10 times for each data point.

rithms on this dataset is also strengthened by the "decade from now" framing of our proposition, as the total number of patents in the United States has grown increasingly rapidly over time. In a decade, the size of this dataset will be notably larger, and thus even harder to compute by direct calculation.

On the quantum side, the HHL algorithm for solving linear systems has been very promising, but currently has some implementation concerns. Most notably, the circuit depth of the data-loading algorithm used in an early practical implementation of HHL is exponential in the number of qubits [6]. It is apparent that the quantum-inspired algorithms are more closely within reach than HHL on a quantum computer for any practical use case.

B. Evaluating the performance of quantum-inspired Grover's algorithm

As for quantum-inspired Grover's algorithm, the argument in favor relies on showing that for at least one problem, the algorithm will perform better than Grover's algorithm as well as any known classical algorithm.

The problem for Grover's algorithm is that despite having a quadratic asymptotic speedup compared to QiGA in the worst case, it has a worse prefactor. This is due to the fact that Grover's algorithm requires $O(2^{n/2})$ calls to the oracle while QiGA requires only one. The actual cost of executing the oracle, which Grover's algorithm traditionally assumes is a black box, creates a larger prefactor for Grover's algorithm than for QiGA.

The claim in Ref. [2] is that there is only a small class of problems for which QiGA is less effective than Grover's algorithm, and they all require extremely large problem instances (a search space of $2^{70} > 10^{21}$ at minimum) as well as high entanglement, resulting in a high bond dimension.

While it is noted in Ref. [2] that more efficient classical algorithms exist to solve the 3-SAT problem, such as Schöning's algorithm, which scales as $O((4/3)^n)$, it is clear that for some instances of 3-SAT, QiGA can be highly efficient. Furthermore, when considering the general k-SAT problem, Schöning's algorithm scales as $O((2(1-1/k))^n)$ [8], so QiGA could potentially be more useful for a larger class of larger k-SAT instances.

We conclude our argument in favor of the proposition with one alternative potential application of QiGA. Consider the subset sum problem, which asks whether or not there is a solution to

$$\sum_{i=1}^{n} a_i x_i = M, \quad \forall i, x_i \in \{0, 1\}, \tag{12}$$

given positive integers $\boldsymbol{a} = (a_1, \ldots, a_n)$ and M. The problem is NP-complete. A quantum oracle can be produced relatively efficiently, using n qubits and n classical "shadow registers", as well as using only CNOT and Toffoli gates [9]. If we can construct this oracle efficiently, subset sum will be a suitable potential candidate for a speedup using QiGA, scaling with a relatively small prefactor (but still exponentially, since it is NP-complete). Crucially, the most efficient current classical algorithm scales as $O(2^{0.283n})$, while a quantum algorithm that relies solely on Grover search has been formulated that can run in $O(2^{0.236n})$ [10]. This means that if QiGA can scale similarly to Grover's in this case, it will be optimal for solving the subset sum problem.

IV. THE CASE AGAINST

While quantum-inspired classical algorithms offer a powerful framework for dequantizing quantum advantages, there are serious obstacles to their becoming the default choice for realistic industrial workloads. In the setting of Ref. [1], the main issues are: (i) the polylogarithmic dependence on matrix dimensions is bought at the price of very steep polynomial factors in rank, condition number, and accuracy; and (ii) even in the most favorable numerical experiments, speedups appear only in a narrow regime of extremely low rank and small condition number. For QiGA [2], efficiency is tightly constrained by entanglement and circuit treewidth, so that the tensor network approach is fast precisely on instances that are already easy for conventional algorithms.

A. Limitations of quantum-inspired SVD

Throughout this section, k denotes the target rank of the low-rank approximation and κ the condition number of A, i.e. the ratio between the largest and smallest nonzero singular values. Intuitively, small k means that the data effectively live in a low-dimensional subspace, and small κ means that solving $A\boldsymbol{x} = \boldsymbol{b}$ is numerically stable.

Both classical and quantum-inspired solvers can start from the same FKV sketch of A. After computing approximate singular values $\tilde{\sigma}_\ell$ and vectors, a classical low-rank solver can explicitly reconstruct the solution vector $\boldsymbol{x}$ in time $O(kn)$, linear in n for fixed k. The quantum-inspired algorithm instead avoids this linear dependence on the ambient dimension by sampling entries of $\boldsymbol{x}$ (via coefficient estimation and rejection sampling), achieving a running time of

$$T_{\mathrm{LS}} = \tilde{O}\big(\kappa^{16} k^6 \|A\|_F^6/\varepsilon^6\big) \tag{13}$$

for linear systems, and a similarly high-degree polynomial in k and $1/\varepsilon$ for recommendation systems [1]. In other words, the price of polylogarithmic dependence on m and n is a harsh sensitivity to rank, conditioning, and target accuracy.

Ref. [1] explicitly emphasizes that, for practical matrix sizes, the direct $O(kn)$ reconstruction "can be done extremely fast even for problems of large size" and that their sampling-based approach only becomes competitive in the regime of very large, very low-rank, and well-conditioned matrices. Outside that sweet spot, the sampling overhead dominates.

This picture is borne out by the random-matrix experiments summarized in Fig. 3. In the most favorable case $m = 40{,}000$, $n = 20{,}000$, $k = \kappa = 5$, and carefully tuned sketch parameters, the quantum-inspired solver achieves a modest wall-clock speedup over direct SVD with a relative error in x of about 9%. However, as k and κ are increased away from this tuned setting, the relative error grows rapidly and approaches $O(1)$ already for $k \approx 50$ and $\kappa \sim 10^2$. The regime in which the method is both accurate and faster than direct SVD is therefore extremely narrow.

The situation becomes even less favorable on real data. On the MovieLens 100K recommendation dataset, the user–item matrix has full rank $k = 611$ and condition number $\kappa \approx 181$. Even when the algorithm is run in a low-rank mode with $k = 10$, the recovered rating vector has relative error $\eta_x \approx 0.71$, and the quantum-inspired pipeline is almost two orders of magnitude slower than a straightforward SVD-based solver on the same hardware [1].

One might hope that patent latent semantic indexing (LSI) provides a more hospitable testbed: the term–document matrices considered in Ref. [7] can be truncated to about $k \approx 80$ singular values while preserving retrieval quality, and thus look closer to the "explicitly low-rank" regime. But the same study finds that LSI only modestly improves over the simpler Vector Space Model (VSM), with typical gains of order 5% and some classes where LSI slightly degrades performance [7]. Combined with the strong polynomial dependence on k, κ, and $1/\varepsilon$, this leaves very little practical headroom: by the time the quantum-inspired pipeline is accurate enough to compete with VSM, its sampling overhead wipes out any potential runtime advantage. In short, for realistic low-rank information-retrieval tasks, the parameter regime in which quantum-inspired SVD is attractive does not seem to overlap with the regime where LSI-style methods are empirically useful.

B. Limitations of quantum-inspired Grover search

The overall cost of QiGA is controlled by the maximum bond dimension $\chi_{\max}$ of the matrix product state representing the intermediate Grover states. In the tensor network implementation of Ref. [2], the runtime scales as

$$T_{\mathrm{QIGA}} = \mathrm{poly}(n)\, \chi_{\max}^3, \tag{14}$$

up to lower order factors. Since $\chi_{\max}$ is directly tied to the bipartite entanglement generated by the oracle circuit, QiGA is efficient only when the Grover oracle admits a low-entanglement, effectively one-dimensional tensor network representation. In contrast, for circuits that generate volume-law entanglement, $\chi_{\max}$ grows exponentially and QiGA inherits the same exponential scaling as generic tensor network simulations.

The random 3-SAT experiments in Ref. [2] illustrate this behavior concretely. Near the satisfiability phase transition, the clause graph is highly connected and has large treewidth, and the observed bond dimensions for instances with $n \approx 30$–40 variables are already in the range $\chi_{\max} \sim 10^3$–10^4. As a result, QiGA runtimes grow from seconds to hours on these tiny instances, even though state-of-the-art classical SAT solvers routinely solve random 3-SAT at $n \approx 40$ in well under a second. In other words, the generic "hard" regime for 3-SAT from the SAT-solver literature coincides with the regime where $\chi_{\max}$ becomes large and QiGA ceases to be competitive.

V. THE CLASS DISCUSSION

We presented our research to our Physics 536 (Quantum Computing) class on December 5, 2025. After our presentation, we noted two major points of disagreement in our rebuttal back-and-forth section: the viability of the asymptotic scaling of the FKV algorithm and coefficient estimation steps, and the viability of a general QiGA tensor network compared to a more specific tensor network construction for each problem.

The quantum-inspired algorithms for linear algebra and recommendation systems take polynomial scaling in the rank k and condition number κ in exchange for avoiding linear scaling in the matrix size m or n. In the context of coefficient estimation, this polynomial scaling can often be worse than a direct reconstruction of the solution vector $\boldsymbol{x}$, and the fact that FKV generally scales as $O(k^6/\varepsilon^6)$ already heavily restricts the problem instances to very low-rank cases [1]. We agreed that the problem could only be industrially viable for large supercomputers for a very limited set of problems, but we disagreed

whether or not these problems would constitute real "industrially default" viability.

As for QiGA, we agreed that there is more research required to measure the entanglement of different oracle constructions (such as subset sum) and better understand the viability of the algorithm for certain problems to see whether or not they are optimal and could constitute practical industrial usage. However, we disagreed on whether a tensor network could always be constructed more specific to the problem structure, such as is the case for the 3-SAT problem focused on in Ref. [2].

While the class was split in their vote on the proposition, more students voted against the proposition than in favor, indicating that they were not convinced that either of these algorithms could constitute widespread industrial usage by 2035.

VI. CONCLUSIONS

While quantum-inspired algorithms probably won't be widely used in an industrial context over established classical algorithms, we believe that they are still formidable classical algorithms as well as being informative in the context of comparison to quantum algorithms.

Their generality makes them interesting to examine for both making connections between classical algorithms that seem very distinct while also functioning as checks on the potential power of quantum algorithms. They are a reason to study quantum algorithms separate from the practicality of actually running quantum algorithms.

We greatly enjoyed the experience of researching these two algorithms for our quantum computing class!

[1] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, Quantum-inspired algorithms in practice, Quantum **4**, 307 (2020).

[2] E. Stoudenmire and X. Waintal, Opening the black box inside grover's algorithm, Physical Review X **14**, 10.1103/physrevx.14.041029 (2024).

[3] A. Frieze, R. Kannan, and S. Vempala, Fast monte-carlo algorithms for finding low-rank approximations, J. ACM **51**, 1025–1041 (2004).

[4] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19 (ACM, 2019) p. 217–228.

[5] A. J. Ferris and G. Vidal, Perfect sampling with unitary tensor networks, Phys. Rev. B **85**, 165146 (2012).

[6] M. Zheng, C. Liu, S. Stein, X. Li, J. Mülmenstädt, Y. Chen, and A. Li, An early investigation of the hhl quantum linear solver for scientific applications, Algorithms **18**, 10.3390/a18080491 (2025).

[7] A. Moldovan, R. I. Boţ, and G. Wanka, Latent semantic indexing for patent documents, International Journal of Applied Mathematics and Computer Science **15**, 551 (2005).

[8] T. Schoning, A probabilistic algorithm for k-sat and constraint satisfaction problems, in *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)* (1999) pp. 410–414.

[9] A. Benoit, S. Schwartz, and R. K. Cytron, Optimization of a quantum subset sum oracle (2024), arXiv:2410.01775 [cs.ET].

[10] X. Bonnetain, R. Bricout, A. Schrottenloher, and Y. Shen, Improved classical and quantum algorithms for subset-sum, Cryptology ePrint Archive, Paper 2020/168 (2020).