Implementation of Trained Factorization Machine Recommendation System on Quantum Annealer

Chen-Yu Liu,[1,2,*] Hsin-Yu Wang,[3,4,†] Pei-Yen Liao,[4,‡] Ching-Jui Lai,[5,§] and Min-Hsiu Hsieh[1,¶]

[1] *Hon Hai Quantum Computing Research Center, Taipei, Taiwan*
[2] *Graduate Institute of Applied Physics, National Taiwan University, Taipei, Taiwan*
[3] *Department of Finance, National Taiwan University, Taipei, Taiwan*
[4] *Department of Mathematics, National Taiwan University, Taipei, Taiwan*
[5] *Department of Mathematics, National Cheng Kung University, Tainan, Taiwan*

(Dated: October 25, 2022)

Factorization Machine (FM) is the most commonly used model to build a recommendation system since it can incorporate side information to improve performance. However, producing item suggestions for a given user with a trained FM is time-consuming. It requires a run-time of $O((N_m \log N_m)^2)$, where N_m is the number of items in the dataset. To address this problem, we propose a quadratic unconstrained binary optimization (QUBO) scheme to combine with FM and apply quantum annealing (QA) computation. Compared to classical methods, this hybrid algorithm provides a faster than quadratic speedup in finding good user suggestions. We then demonstrate the aforementioned computational advantage on current NISQ hardware by experimenting with a real example on a D-Wave annealer.

I. INTRODUCTION

Recommendation systems are widely used to predict the rating or preference of items for arbitrary users, such as suggesting books or movies, and also for many other business applications [1–4]. A common principle for constructing a recommendation system assumes that users prefer similar things based on past behavior. Amongst several approaches, the objective of Matrix Factorization (MF) is to find the factor matrices of historical data [5, 6]. Together with MF, other improvements such as SVD++ [7], pairwise interaction tensor factorization (PITF) [8], factorizing personalized Markov chains (FPMC) [9] and Monte Carlo on bipartite graphs [10] have also been comprehensively studied. These methods introduce specialized models to treat specific tasks, but with the trade-off of losing generality.

To fix this, the Factorization Machine (FM) [11] is proposed as a more general supervised learning method, unifying MF, SVD++, PITF, and FPMC. With the capability to input arbitrary real-valued feature vectors to the model, including side information (e.g., gender and age), FM can be used for regression, classification, and ranking tasks. Moreover, it can be trained with linear complexity and accurately estimate model parameters from a sparse dataset, which makes FM highly competent for analyzing large datasets. Thus, it is more natural and adequate to utilize FM for real-world applications than the other methods mentioned earlier.

In recent years, quantum computing has become one of the most popular domains, which aims to benefit from the superposition nature of quantum states. Quantum Machine Learning (QML) employs parameterized quantum circuits as a neural network, which can then be applied to different classes of machine learning, such as Quantum Neural Network (QNN) [12–14], Quantum Convolutional Neural Network (QCNN) [15–17], Quantum Reinforcement Learning (QRL) [18–20], and Quantum Generative Adversarial Neural Network (QGAN) [21–25]. Besides the structural exploration, several works [26–34] have also investigated the learnability and trainability of QNN. Although the data encoding to a Hilbert space may give potential computational advantages, at the current stage of NISQ era, only low-dimensional or small sample problems can be implemented on real-world hardware. Thus it is hard to justify the advantage experimentally, and this needs to be further studied [35].

The concept of a Quantum Recommendation System (QRS) has also been proposed [36], for which the idea is to sample an approximation of the preference matrix by quantum singular value estimation and quantum projection. In that framework, QRS takes $O(\text{poly}(k)\text{polylog}(mn))$ running time with k the rank of the approximation and mn the matrix dimension, while classically reconstructing an approximation of the preference matrix requires polynomial time mn. However, a classical analogue of QRS, called Quantum-Inspired Recommendation System (QIRS), provides $O(\text{poly}(k)\log(mn))$ running time [37], which closes the gap between the classical and quantum methods. Besides gate-based quantum computing, it is possible to apply quantum annealing for feature selection [38] and use these features to train a classical ItemKNN content-based model [39]. This hybrid QPU solver shows good scalability for larger instances but may not have a significant advantage in the running time.

The performance of QRS or QIRS depends on a good low-rank approximation to the preference matrix [36, 37]. Moreover, since these algorithms do not incorporate side

* cyliuphys@gapp.nthu.edu.tw
† b07703009@ntu.edu.tw
‡ b08705006@ntu.edu.tw
§ cjlai72@mail.ncku.edu.tw
¶ min-hsiu.hsieh@foxconn.com

information, they may not be competitive with FM. Thus in the recommendation system domain, an approach to both utilize the side information and attain potential speedup is a crucial target for practical quantum computing applications.

In this work, we formulate a hybrid recommendation system by incorporating a classically trained FM with a quadratic unconstrained binary optimization (QUBO) scheme to be solved quantumly. Applying Quantum Annealing (QA) computation enables us to achieve a quadratic speedup in the suggesting process compared to classical methods. Our algorithm will soon provide a computational advantage from the prospects of today's developing NISQ hardware, such as a D-Wave annealer, for practical problems.

A. Main Results

Our main results in both theoretical and experimental speedup of a QA-assisted FM recommendation system are summarized as follows:

- Suggestion process as energy minimizing task: We propose a QUBO scheme for the suggestion process of the FM recommendation system. The energy minimization task of the corresponding Ising problem is equivalent to finding the highest score (rating) candidates in the recommendation system.
- Speedup by Quantum Annealing: The computational complexity that directly searches the entire dataset is $O((N_m \log N_m)^2)$, where N_m is the number of candidates in the dataset. In contrast, the time complexity for our QA-assisted hybrid algorithm is $O(e^{\sqrt{\log N_m}} \log N_m)$, which provides a faster than quadratic speedup than direct search. Furthermore, the computational scaling behavior from the experimental results obtained from the classical computer and D-Wave's quantum annealer is compatible with our theoretical results.
- Scalability: The capability of solving fully-connected size-N_p Ising problems of the most advanced D-Wave Advantage 4.1 system is $N_p = 145$. At the same time, the corresponding QUBO size of suggesting N_m candidates is $\lceil \log_2 N_m \rceil$ in our formulation. Hence, theoretically, the solvable problem size N_m scales up to $2^{145} \approx 10^{43}$ in today's quantum hardware.

B. Related Work

Several works apply the QUBO scheme to the machine learning study. Date *et al.* [40] proposes QUBO formulations of linear regression, support vector machine (SVM), and balanced k-means clustering, where the required qubit number usually scales as $O(N_{\text{data}}^2)$, with N_{data} the number of the data points of the training data. However, N_{data} could go easily beyond 10^3 in most machine learning applications, which is higher than the capability of the most advanced quantum system for solving an Ising problem: $N_{\text{data}} = 145$ from D-Wave [41]. One can transform an FM into a QUBO problem by binary encoding with a size equivalent to the length of the input vector. The works [42–45] have applied this idea in structural design problems to maximize the acquisition function associated to an FM. In this manner, the size of the QUBO problem is usually within a range that today's hardware could reach. These examples open the possibility of applying such a QUBO-FM scheme to any FM-related research.

II. PRELIMINARIES

In this preliminary section, we review the key ingredients of our work: FM, QUBO, and Quantum Annealing (QA). FM is the most commonly used model in recommendation systems and machine learning. The QUBO scheme can apply to various combinatorial optimization problems, and QA is an effective optimization method for solving QUBO problems.

A. Factorization Machine

A factorization machine (FM) is a supervised learning algorithm that can be used for classification, regression, and ranking tasks [11]. Here we consider FM with degree $D = 2$, which involves only single and pairwise interactions of items. The model equation for an FM of degree $D = 2$ with variables

$$\vec{x} = (x_1, \ldots, x_d)$$

is defined as:

$$y_{\text{fm}}(\vec{x}) := w_0 + \vec{w} \cdot \vec{x} + \sum_{i<j}^{d} v_i^T v_j x_i x_j, \tag{1}$$

where $w_0 \in \mathbb{R}$ is the global bias, $\vec{w} = (w_1, \ldots, w_d) \in \mathbb{R}^d$ refers to the weights of each variables and

$$\mathbf{V} = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \ldots & v_d \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{k \times d} \tag{2}$$

denotes the feature embeddings with k the dimensionality of latent factors. The term $v_i^T v_j$ represents the interaction between the i-th and j-th terms.

We can use FM to construct a recommendation system: Separate the variables $\vec{x}$ into $x_i = u_i$ for $1 \le i \le n_u$ and $x_{i+n_u} = m_i$ for $1 \le i \le n_m$, so that the vector $\vec{u} = (u_1, \ldots, u_{n_u})$ denotes the information of a user and $\vec{m} = (m_1, \ldots, m_{n_m})$ represents the information of an item to be recommended, with the dimension relation $n_u + n_m =$

d, this then turns $y_{\text{fm}}(\vec{x})$ into a predictor that estimates the score or rating for such a $(\vec{u}, \vec{m})$ pair.

FM model is known in the original FM paper [11] to have linear complexity, the capability of parameter estimation under sparse data, and the ability to work with any real-valued feature vector. As a result, one can use an FM model as a more general predictor than other state-of-art factorization models that only work on restricted input data and require individual task analysis. Furthermore, it has been shown in the same article that FM can mimic biased MF, SVD++, PITF, and FPMC, which indicates that FM is user-friendly for non-experts wanting to work with factorization models.

One can describe a typical use case of FM as follows:

1. Prepare a set of data with user information, item information, and ratings (or scores) of the items produced by the user.

2. With such a dataset, w_0, w, and v can then be learned by optimization algorithms, such as stochastic gradient descent (SGD), alternating least-squares (ALS) optimization, as well as Bayesian inference using Markov Chain Monte Carlo (MCMC) [46]. This step is also called the training process.

3. After training an FM, the predictor $y_{\text{fm}}(\vec{x})$ can estimate the ratings of every item in the dataset with given user information.

With a trained FM, we perform the suggestion process of a given user in the following manner. First, with N_m items in the dataset, each item is input to the FM to produce the corresponding rating. Next, we sort the obtained N_m ratings and provide the list of the top k_s suggestions to the user, where k_s is the pre-determined number of suggestions.

In the rest of this work, the suggestion method that calculates all ratings directly and then sorts is called the "direct method".

For a trained FM with k the dimensionality of the factorization, the complexity of finding the predictor $y_{\text{fm}}(\vec{x})|_{\vec{u}=\vec{u}^0}$ for a given user $\vec{u}^0$ is $O(kn_m)$ from eq. (9) and [11]. The complexity of sorting N_m items is classically known as $O(N_m \log N_m)$. Hence to sort N_m ratings for the user $\vec{u}^0$ with a trained FM, with size-n_m vectors representing these items, requires the complexity $O(N_m \times kn_m \times N_m \log N_m)$. Since $k = O(1)$, the overall complexity of the suggestion process for a fixed user in the direct method is $O(n_m N_m^2 \log N_m)$.

Note that if the vectors $\vec{m}$ for encoding N_m items are in binary form and denote $\log \equiv \log_2$, then $n_m \approx \log N_m$. In this case, the complexity of the suggestion process becomes $O((N_m \log N_m)^2)$.

B. Quadratic Unconstrained Binary Optimization

Quadratic unconstrained binary optimization (QUBO) is an NP-hard combinatorial problem that can apply to the traveler salesman problem, finance portfolio optimization, max-cut problem, and even machine learning [47]. With binary variables $x_i \in \{0, 1\}$, linear weights $w_i \in \mathbb{R}$, and coefficients $W_{ij} \in \mathbb{R}$, the equation of a QUBO problem is:

$$y_{\text{QUBO}}(\vec{x}) := \sum_{i=1}^{n} w_i x_i + \sum_{i<j}^{n} W_{ij} x_i x_j. \tag{3}$$

The solution to a QUBO problem is a binary vector $\vec{x}^*$ that minimizes the objective value y_{QUBO}:

$$\vec{x}^* = \arg\min_{\vec{x}} y_{\text{QUBO}}(\vec{x}). \tag{4}$$

Note that one can transform a minimization problem into a maximization problem by adding a minus sign to the objective value. Moreover, it is possible to map a QUBO problem into an Ising problem [48] with the relation $\sigma_i^z = 2x_i - 1$. In particular, a QUBO problem is computationally equivalent to the following form :

$$H_p = \sum_{i=1}^{n} h_i \sigma_i^z + \sum_{i<j}^{n} J_{ij} \sigma_i^z \sigma_j^z, \tag{5}$$

where $h_i \in \mathbb{R}$ and $J_{ij} \in \mathbb{R}$ are the corresponding external field and coupling terms, and σ^z is the Pauli-z operator. Thus, we can solve a QUBO problem by solving its corresponding Ising form. This approach is achievable by energy minimization through adiabatic quantum computation, such as quantum annealing.

C. Quantum Annealing

Quantum Annealing (QA), formulated in its current form by T. Kadowaki and H. Nishimori [49], is a metaheuristic approach capable of solving combinatorial optimization problems by utilizing the quantum mechanical nature of the associated physical systems. The QA process evolves from the ground state $|\psi_i\rangle$ of an initial Hamiltonian H_i to the ground state $|\psi_p\rangle$ of the problem Hamiltonian H_p (eq. (5)), where presumably one can easily prepare $(H_i, |\psi_i\rangle)$. Mathematically, this adiabatic evolution can be described by the Hamiltonian in the following form, with $s(t)$ evolving from 1 to 0 during a QA process:

$$H(t) = s(t)H_i + (1 - s(t))H_p. \tag{6}$$

For example, it is set to be $H_i \propto \sum_{i=1}^{n} \sigma_i^x$ in D-Wave's implementation [50]. With a proper annealing schedule $s(t)$, probability of finding the ground state of H_p is close to 1 [51]. Thus, QA can search for the solution to a QUBO problem by studying the corresponding Ising form (eq. (5)). This approach has a wide range of applications for different problems mentioned in Sec. II B [47].

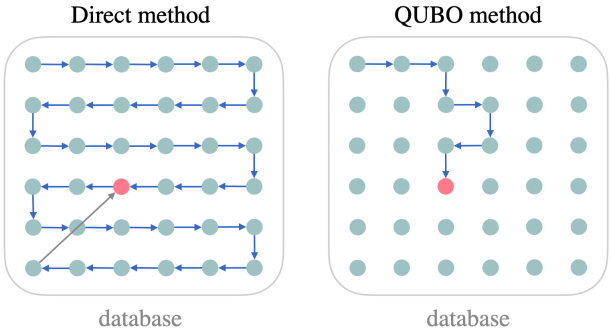

FIG. 1. Schematic diagram. The direct method needs to predict the rating for every data point in the database and return the highest rating candidate at the end (red dot). In contrast, the QUBO method considers the database as the energy landscape of an energy minimization problem, which only calculates the data points along the optimization trajectory. A specific optimization method can significantly reduce the computational complexity (quantum annealing in our case).

III. TRAINED FACTORIZATION MACHINE AS QUBO

In this section, we first derive the QUBO formulation of the suggestion process with a given trained FM and a fixed user. Then, we estimate the computation complexity of applying QA to the associated optimization problem. The key component here is that we will specialize the d-dimension input vector $\vec{x} = (\vec{u}, \vec{m})$ in eq. (1) to be in the *binary form* where $\vec{u} \in \{0,1\}^{n_u}$ encodes a user and $\vec{m} \in \{0,1\}^{n_m}$ an item in the dataset, with the dimension relation $n_u + n_m = d$.

Consider an FM recommendation system modeled by eq. (1), where the training process described in Sec. II A determines the global bias $w_0 \in \mathbb{R}$, the weight $\vec{w} \in \mathbb{R}^d$ and the feature embedding $\mathbf{V} \in \mathbb{R}^{k\times d}$. When assuming the input vector $\vec{x}$ to be binary, we view eq. (1) as a QUBO problem equation eq. (3) with $\vec{w}$ remaining the same and $W_{ij} = v_i^T v_j$. Note that eliminating the bias term w_0 does not affect the underlying optimization problem. Given the meaning of $\vec{x}$ in FM to be a pair of user and item data, and y_{fm} the corresponding rating (or score), by introducing a negative sign to the associated QUBO problem:

$$-y_{\text{fm}}(\vec{x}) = y_{\text{QUBO}}(\vec{x}) = \sum_{i=1}^{d} w_i x_i + \sum_{i<j}^{d} W_{ij} x_i x_j, \quad (7)$$

the solution $\vec{x}$ now becomes the highest rating pair for the resulting minimization problem.

To formulate the suggestion process for a picked user represented by the binary vector $\vec{u}^0$, we fix the user part $\vec{u} = \vec{u}^0$ in the input vector $\vec{x} = (\vec{u}, \vec{m})$, which reduces the dimension of the corresponding QUBO from d to n_m:

$$\begin{aligned} y_{\text{QUBO}}(\vec{x})|_{\vec{u}=\vec{u}^0} &= \sum_{i=1}^{d} w_i x_i + \sum_{i<j}^{d} W_{ij} x_i x_j \\ &= \sum_{i=1}^{n_u} w_i u_i^0 + \sum_{i=1}^{n_m} w_{n_u+i} m_i \\ &\quad + \sum_{i<j}^{n_u} W_{ij} u_i^0 u_j^0 + \cdots + \sum_{i<j}^{n_m} W_{i+n_u,j+n_u} m_i m_j. \end{aligned} \quad (8)$$

By summing over like terms, we further reduce this equation to

$$\begin{aligned} &= \sum_{i=1}^{n_m} \tilde{w}_i m_i + \sum_{i<j}^{n_m} \tilde{W}_{ij} m_i m_j + \text{offset} \\ &\equiv \sum_{i=1}^{n_m} \tilde{w}_i m_i + \sum_{i<j}^{n_m} \tilde{W}_{ij} m_i m_j, \end{aligned} \quad (9)$$

where the "offset" term in the last step is neglected as it does not affect the underlying optimization problem. Finally, we can map eq. (9) into an n_m-dimensional Ising problem as discussed in Sec. II B:

$$y_{\text{QUBO}}(\vec{x})|_{\vec{u}=\vec{u}^0} = \sum_{i=1}^{n_m} \tilde{h}_i \sigma_i + \sum_{i<j}^{n_m} \tilde{J}_{ij} \sigma_i \sigma_j. \quad (10)$$

By calculating the ground states and possibly some low-lying excited states of the Ising problem with QA or other algorithms, we can transform the quest of producing high-rating recommendations for the trained FM into the task of finding low-lying energy states for the corresponding Ising problem.

It has been observed in the experiment that the computational complexity of calculating an N-dimensional Ising system ground state by quantum annealing is $O(e^{\sqrt{N}})$ [52, 53]. With a given trained FM and binary encoded dataset, the following proposition provides the computational complexity of the suggestion process with a fixed user and N_m items by utilizing quantum annealing to the Ising problem eq. (10).

Proposition III.1 *For the suggestion process with a fixed user and N_m items in the dataset, the hybrid algorithm of transforming a trained FM into QUBO and then applying QA computation to the associated Ising problem has the run-time complexity $O(e^{\sqrt{\log N_m}} \log N_m)$.*

In particular, the proposed hybrid algorithm has a quadratic speedup compared to the direct method discussed in Sec. II A.

The last statement in the above proposition is justified as the following: If N_m is large, then the above-mentioned complexity has the magnitude

$$e^{\sqrt{\log N_m}} \cdot \log N_m \approx \log N_m \cdot 2^{1.442\sqrt{\log N_m}},$$

which is (much) smaller than the square root $\log N_m \cdot 2^{\log N_m}$ of the complexity function of the direct method.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

We use the well-known MovieLens-20M dataset [54] to demonstrate the effectiveness and behaviors of our hybrid algorithm, with $\vec{u}$ and $\vec{m}$ the binary encoding of userID and movieID, and $y_{\text{fm}}(\vec{x})$ the prediction of rating. Note that even though FM can utilize side information of users and movies, here we only use userID and movieID to train an FM. The dataset MovieLens-20M contains 20 million ratings of 27000 movies by 138000 users, and we use different fractions of this dataset and up to 6 million ratings to observe behaviors of our method.

A. Set Up

In our experiment, we first train an FM model with latent factors of dimension $k = 200$ and then transform it into the corresponding Ising problem for the suggestion process.

However, there is an issue with binary encoding: the dimension of the solution space in quantum computing is typically more than the number of items (all possible movies for us) from the dataset. For the direct method, the solution space of the suggestion process is the number N_m of all the movies, while the dimension of the solution space for the QUBO method is $2^{\lceil \log_2 N_m \rceil}$. In this case, QUBO method may consider $2^{\lceil \log_2 N_m \rceil} - N_m$ binary vectors that is not in the movie list as solutions. To deal with this issue, we map all the computational basis vectors of the solution Hilbert space to our movie list (item dataset) by encoding the $2^{\lceil \log_2 N_m \rceil} - N_m$ higher-rating movies (items) in the dataset to 2 different binary vectors. Thus, any solution sampled by quantum annealing will produce one of the existing movies.

The FM in this case is trained by 6 million ratings for 41305 users and 21011 movies, so that the lengths for the user and movie part of the input vector are respectively $n_u = \lceil \log_2 41305 \rceil = 16$ and $n_m = \lceil \log_2 21011 \rceil = 15$. In particular, the QUBO/Ising problem size of the suggestion process for a fixed user is 15. We use the quantum annealer device D-Wave DW_2000Q_6, which has 2000 qubits and 6000 couplers.

B. Solution Quality

We use the *overlapping rate* (in percentage) as the standard of our search results. Here the overlapping rate in the top-k_s experiment is defined as the proportion of QA-captured overlapped movies in the top-k_s list from the direct method. Fig. 2(a) shows the overlapping rate of QA movie recommendation for both 100 measurement shots and 4000 measurement shots.

In Fig. 2(a), each data point represents the average of QA suggesting results from 100 randomly picked users, while a vertical strip represents the overlapping rate of each QA 4000/100-shot results for $k_s \in \{10, 30, 50\}$. We observe that QA captures only small amount of targeting movies from the 100-shot result but relatively more movies from the 4000-shot result.

On average, the overlapping rates are 51.4%, 39.07%, and 32.96% from the top-10, top-30, and top-50 suggestions from the 4000-shot results. The overlapping rate is higher for a smaller k_s, which is consistent with the principle that QA samples more frequently lowly-excited states.

C. Speedup by Quantum Annealing

From Sec. II A and III, the computational complexity of the suggestion process for the direct method and the QA approach are given respectively by $O((N_m \log N_m)^2)$ and $O(e^{\sqrt{\log N_m}} \log N_m)$. By using different fractions of MovieLens-20M dataset as training data, our experiment also serves as a test for the scaling behavior in N_m of both algorithms.

Note that the amount of movies varies from different fractions of dataset. Fig. 2(b) shows the execution time for direct method and QA for $N_m \in \{2090, 8227, 11719, 13950, 16715\}$, where each data point represents the average results of 5 randomly chosen users. We perform QA on the D-Wave DW_2000Q_6 QPU and the direct method on an Apple M1 Max chip with a 10-core CPU. As a result, the experimental data points fit the complexity functions mentioned above with regression factors.

The expected execution time $O(e^{\sqrt{\log N_m}} \log N_m)$ shows that QA is an efficient sampling method for suggesting candidates in the FM recommendation system. With some sacrifice of solution quality (Fig. 2(a)), this could be useful if N_m is extremely large and there is no efficient algorithm in the classical approaches.

D. Scalability

Our experiment includes examples of dataset size $N_{\text{data}} \in \{5 \times 10^3, 10^5, 4 \times 10^5, 10^6, 2 \times 10^6, 6 \times 10^6\}$ with the corresponding QUBO size $n_m \in \{12, 14, 15\}$.

In Fig, 3, we extrapolate the fitting curves in Fig. 2 up to $N_m \sim 10^{49} \sim 2^{163}$ to see the execution time scaling in the extreme case. To visualize the capability of the current D-Wave quantum annealers, we also include the estimation of the corresponding QUBO size upper limits in the graph. For the 5760-qubit D-Wave Advantage 4.1 system, the upper limit for a general Ising problem is around 145 due to the non-fully connectedness of current hardware. For the same reason, this upper limit for the 2000-qubit D-Wave DW_2000Q system is 64 [41].

We can observe that the fitting curve related to QA is dozens of orders less than that of the direct method near the case of the D-Wave upper limit. The "QUBO size" axis of the figure can also be considered as the length

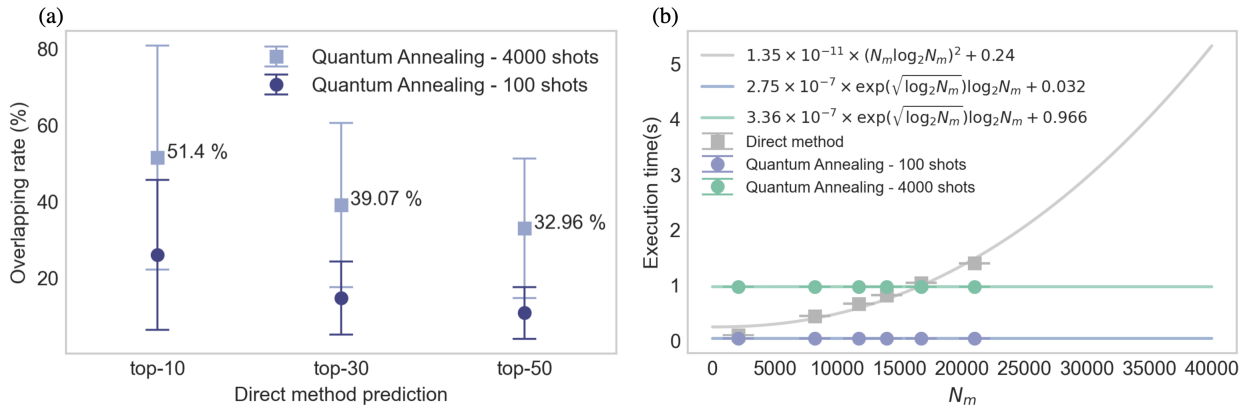

FIG. 2. (a) Overlapping ratings between predictions from direct method and QA for both 100 measurement shots and 4000 measurement shots. The FM is trained by 6 million ratings for 41305 users and 21011 movies ($N_{\text{data}} = 6 \times 10^6, N_u = 41305, N_m = 21011, n_u = 16, n_m = 15$). (b) Execution time for both direct method and QA with $N_{\text{data}} \in \{5 \times 10^3, 10^5, 4 \times 10^5, 10^6, 2 \times 10^6, 6 \times 10^6\}$, $N_m \in \{2090, 8227, 11719, 13950, 16715, 21011\}$ and $n_m \in \{12, 14, 15\}$.

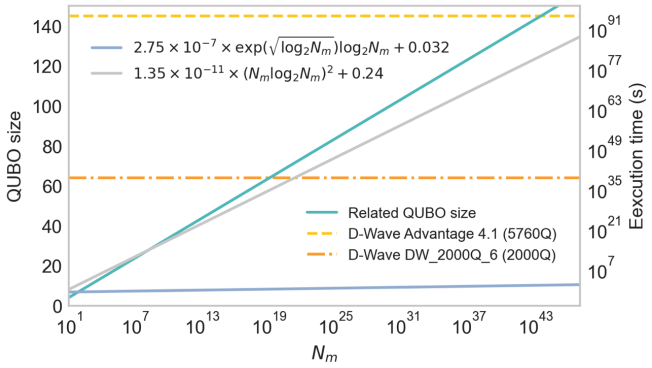

FIG. 3. Scaling behaviors of the execution time for both direct method and QA in the case of N_m from 10^1 to 10^{47}. Corresponding QUBO sizes ($\lceil \log_2 N_m \rceil$) of different N_m's are also plotted, with indications of the problem size upper limits for D-Wave's current hardware.

of the encoded binary vector of the items in the problem formulation and is not necessarily the same as our MovieLens example. In particular, it can include more information besides item identities, as in our case.

E. Effects of Annealing Parameters

Since QA involves several factors that could affect our experiment, to understand our algorithm's properties better, we investigate different tunable parameters of the QA process to see the behavior of the results. We select four tunable parameters to investigate: "Annealing time", "Programming thermalization", "Readout thermalization" and "Shots". According to the solver parameters document from D-Wave [55]: "Annealing time" is the time duration of the quantum annealing; "Programming thermalization" is the time to wait after programming the quantum annealer for it to cool back to a base temperature; "Readout thermalization" is the time to wait after each state is read from the quantum annealer for it to cool back to the base temperature; "Shots" is the number of states to read from the solver. We use the number of overlapping movies between top-100 suggestions from the direct method and QA to analyze the performance.

Fig. 4(a) investigates different parameters of programming thermalization and annealing time, where the number of overlapping movies shows no significant difference among these variations of parameters. One can also observe similar behavior in Fig. 4(b): the readout thermalization and annealing time could result in the thermal noise in the system, which is in the same order of magnitude as the thermal variation before and after the waiting time. For annealing time, these results show that the default value 20 μs is already sufficient for our QA process. Thus no improvement would be found with larger values. In Fig. 4(c), it is no surprise that results with more shots perform better since we have more chances to hit the high score candidates in these cases.

V. CONCLUSION AND FUTURE WORK

In this work, by transforming a trained FM from user dataset into a QUBO formulation, we construct a hybrid recommendation system based on solving the associated Ising problem via quantum annealing.

Theoretically, we first derive the QUBO formulation for the suggestion process and then prove that the computational complexity has an faster than quadratic speedup improvement:

$$N_m^2 \log N_m \to e^{\sqrt{\log N_m}}$$

compared to the classic direct method.

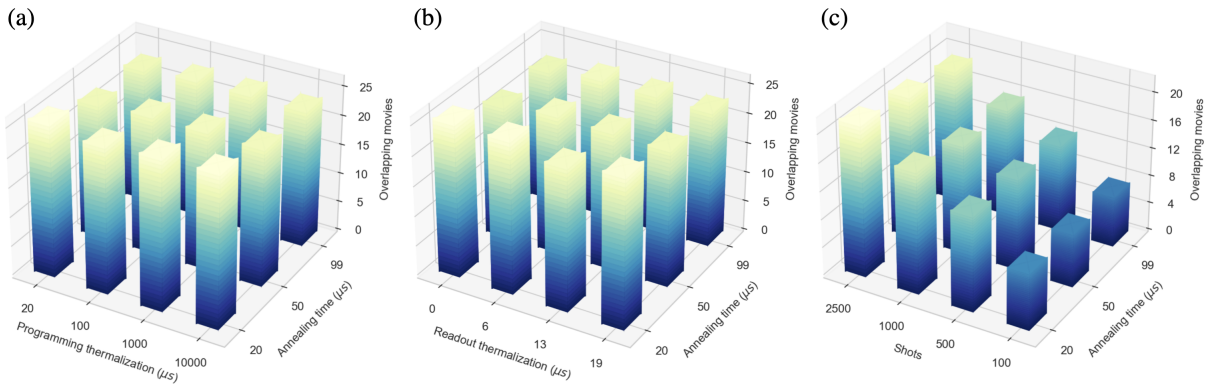

FIG. 4. The number of overlapping movies between top-100 suggestions from the direct method and QA for different annealing parameters. (a) Different programming thermalization and annealing time with measurement shots = 2500 and readout thermalization = 0 μs. (b) Different readout thermalization and annealing time with measurement shots = 2500 and programming thermalization = 1000 μs. (c) Different measurement shots and annealing time with readout thermalization = 0 μs and programming thermalization = 1000 μs.

Experimentally, we apply our hybrid algorithm to the MovieLens-20M dataset to demonstrate the applicability of current NISQ hardware and analyze the solution quality. By encoding the whole solution space of QUBO to the movie list, we use the D-Wave quantum annealer to solve the corresponding optimization problem of the suggestion process. Our experiment shows that the top-10 suggestion has 51.4% average overlapping rate with that from the direct method, and has the scaling behavior of the execution time matching our theoretical results.

Our future work could be searching for other use cases of FM and studying whether the quantum annealing approach is advantageous in such cases.

VI. ACKNOWLEDGMENTS

Hsin-Yu Wang, Pei-Yen Liao and Ching-Jui Lai acknowledge the funding support from the Ministry of Education in Taiwan for developing an innovative training scheme to bring intelligent students to frontier research. We thank Ming-Da Liu and Chiao-Ching Huang from the Data Management department of Cathay Life Insurance in Taiwan for bringing up this question, and Simon Lin for valuable discussions. Finally, we thank the Center for Quantum Frontiers of Research and Technology, National Cheng Kung University, Taiwan, for supporting our access to D-Wave's QPU through Amazon Braket.

[1] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, Recommender system application developments: A survey, Decision Support Systems **74**, 12 (2015).
[2] Y. Shi, M. Larson, and A. Hanjalic, Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges, ACM Comput. Surv. **47**, 10.1145/2556270 (2014).
[3] J. Tang, X. Hu, and H. Liu, Social recommendation: a review, Social Network Analysis and Mining **3**, 1113 (2013).
[4] X. Yang, Y. Guo, Y. Liu, and H. Steck, A survey of collaborative filtering based social recommender systems, Computer Communications **41**, 1 (2014).
[5] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, edited by F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor (Springer US, Boston, MA, 2011) pp. 1–35.
[6] Y. Koren, R. Bell, and C. Volinsky, Matrix factorization techniques for recommender systems, Computer **42**, 30 (2009).
[7] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08 (Association for Computing Machinery, New York, NY, USA, 2008) p. 426–434.
[8] S. Rendle and L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10 (Association for Computing Machinery, New York, NY, USA, 2010) p. 81–90.

[9] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10 (Association for Computing Machinery, New York, NY, USA, 2010) p. 811–820.

[10] Y. Rong, X. Wen, and H. Cheng, A monte carlo algorithm for cold start recommendation, in *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 (Association for Computing Machinery, New York, NY, USA, 2014) p. 327–336.

[11] S. Rendle, Factorization machines, in *2010 IEEE International Conference on Data Mining* (2010) pp. 995–1000.

[12] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, Quantum Information Processing **13**, 2567 (2014).

[13] M. V. Altaisky, Quantum neural network (2001), arXiv:quant-ph/0107012.

[14] S. C. Kak, Quantum neural computing, Advances in Imaging and Electron Physics **94**, 259 (1995).

[15] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nature Physics **15**, 1273 (2019).

[16] Y. Li, R.-G. Zhou, R. Xu, J. Luo, and W. Hu, A quantum deep convolutional neural network for image recognition, Quantum Science and Technology **5**, 044003 (2020).

[17] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, Quantum convolutional neural networks for high energy physics data analysis, Phys. Rev. Research **4**, 013231 (2022).

[18] D. Dong, C. Chen, H. Li, and T.-J. Tarn, Quantum reinforcement learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **38**, 1207 (2008).

[19] J.-Y. Hsiao, Y. Du, W.-Y. Chiang, M.-H. Hsieh, and H.-S. Goan, Unentangled quantum reinforcement learning agents in the openai gym (2022), arXiv:2203.14348.

[20] V. Dunjko, J. M. Taylor, and H. J. Briegel, Advances in quantum reinforcement learning, in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2017) pp. 282–287.

[21] P.-L. Dallaire-Demers and N. Killoran, Quantum generative adversarial networks, Phys. Rev. A **98**, 012324 (2018).

[22] S. Lloyd and C. Weedbrook, Quantum generative adversarial learning, Phys. Rev. Lett. **121**, 040502 (2018).

[23] J. Tian, X. Sun, Y. Du, S. Zhao, Q. Liu, K. Zhang, W. Yi, W. Huang, C. Wang, X. Wu, M.-H. Hsieh, T. Liu, W. Yang, and D. Tao, Recent advances for quantum neural networks in generative learning (2022), arXiv:2206.03066.

[24] Y. Du, M.-H. Hsieh, and D. Tao, Efficient online quantum generative adversarial learning algorithms with applications (2019), arXiv:1904.09602.

[25] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu, and J.-W. Pan, Experimental quantum generative adversarial networks for image generation, Phys. Rev. Applied **16**, 024051 (2021).

[26] Y. Du, M.-H. Hsieh, T. Liu, S. You, and D. Tao, Learnability of quantum neural networks, PRX Quantum **2**, 040337 (2021).

[27] M. Soltanolkotabi, A. Javanmard, and J. D. Lee, Theoretical insights into the optimization landscape of over-parameterized shallow neural networks, IEEE Transactions on Information Theory **65**, 742 (2019).

[28] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, A grover-search based quantum learning scheme for classification, New Journal of Physics **23**, 023020 (2021).

[29] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, Expressive power of parametrized quantum circuits, Phys. Rev. Research **2**, 033125 (2020).

[30] K. Zhang, M.-H. Hsieh, L. Liu, and D. Tao, Toward trainability of quantum neural networks (2020), arXiv:2011.06258.

[31] K. Zhang, M.-H. Hsieh, L. Liu, and D. Tao, Gaussian initializations help deep variational quantum circuits escape from the barren plateau (2022), arXiv:2203.09376.

[32] K. Zhang, M.-H. Hsieh, L. Liu, and D. Tao, Quantum gram-schmidt processes and their application to efficient state readout for quantum algorithms, Phys. Rev. Research **3**, 043095 (2021).

[33] Y. Du, M.-H. Hsieh, T. Liu, S. You, and D. Tao, Quantum differentially private sparse regression learning, IEEE Transactions on Information Theory **68**, 5217 (2022).

[34] Y. Du, M.-H. Hsieh, T. Liu, D. Tao, and N. Liu, Quantum noise protects quantum classifiers against adversaries, Phys. Rev. Research **3**, 023153 (2021).

[35] R. Zhao and S. Wang, A review of quantum neural networks: Methods, models, dilemma (2021), arXiv:2109.01840.

[36] I. Kerenidis and A. Prakash, Quantum recommendation systems (2016), arXiv:1603.08675.

[37] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (Association for Computing Machinery, New York, NY, USA, 2019) p. 217–228.

[38] R. Nembrini, M. Ferrari Dacrema, and P. Cremonesi, Feature selection for recommender systems with quantum computing, Entropy **23**, 10.3390/e23080970 (2021).

[39] M. Deshpande and G. Karypis, Item-based top-¡i¿n¡/i¿ recommendation algorithms, ACM Trans. Inf. Syst. **22**, 143–177 (2004).

[40] P. Date, D. Arthur, and L. Pusey-Nazzaro, Qubo formulations for training machine learning models, Scientific Reports **11**, 10029 (2021).

[41] D-Wave, Upper limit for an arbitrary ising problem on d-wave, `https://github.com/aws/amazon-braket-examples/blob/main/examples/quantum_annealing/Running_large_problems_using_QBSolv.ipynb` (n.d.).

[42] K. Kitai, J. Guo, S. Ju, S. Tanaka, K. Tsuda, J. Shiomi, and R. Tamura, Designing metamaterials with quantum annealing and factorization machines, Phys. Rev. Research **2**, 013319 (2020).

[43] K. Kitai, J. Guo, S. Ju, S. Tanaka, K. Tsuda, J. Shiomi, and R. Tamura, Designing metamaterials with quantum annealing and factorization machines, Phys. Rev. Research **2**, 013319 (2020).

[44] T. Matsumori, M. Taki, and T. Kadowaki, Application of qubo solver using black-box optimization to structural design for resonance avoidance, Scientific Reports **12**, 12143 (2022).

[45] K. Hatakeyama-Sato, T. Kashikawa, K. Kimura, and K. Oyaizu, Tackling the challenge of a huge materials science search space with quantum-inspired annealing,

Advanced Intelligent Systems **3**, 2000209 (2021).
[46] S. Rendle, Factorization machines with libfm, ACM Trans. Intell. Syst. Technol. **3**, 10.1145/2168752.2168771 (2012).
[47] A. Lucas, Ising formulations of many np problems, Frontiers in Physics **2**, 10.3389/fphy.2014.00005 (2014).
[48] Z. Bian, F. Chudak, W. Macready, A. Roy, R. Sebastiani, and S. Varotti, Solving sat and maxsat with a quantum annealer: Foundations, encodings, and preliminary results (2018), arXiv:1811.02524.
[49] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse ising model, Phys. Rev. E **58**, 5355 (1998).
[50] D-Wave, The hamiltonian and the eigenspectrum, https://docs.dwavesys.com/docs/latest/c_gs_2.html#the-hamiltonian-and-the-eigenspectrum (n.d.).
[51] T. Albash and D. A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys. **90**, 015002 (2018).
[52] S. Mukherjee and B. K. Chakrabarti, Multivariable optimization: Quantum annealing and computation, The European Physical Journal Special Topics **224**, 17 (2015).
[53] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, Evidence for quantum annealing with more than one hundred qubits, Nature Physics **10**, 218 (2014).
[54] Movielens-20m dataset, https://grouplens.org/datasets/movielens/20m/ (2015).
[55] D-Wave, Solver parameters, https://docs.dwavesys.com/docs/latest/c_solver_parameters.html (n.d.).
[56] IBM Quantum, https://quantum-computing.ibm.com (n.d.).
[57] Amazon Braket, https://aws.amazon.com/braket (2022).
[58] D-Wave, https://www.dwavesys.com (2022).
[59] M. Booth and S. P. Reinhardt, D-Wave Technical Report Series **14-1006A-A** (2017).