



A quantum “black box” for entropy calculation

Michal Koren¹ · Oded Koren¹ · Or Peretz¹

Received: 15 December 2022 / Accepted: 11 August 2023
© The Author(s) 2023

Abstract

A significant part of global quantum computing research has been conducted based on quantum mechanics, which can now be used with quantum computers. However, designing a quantum algorithm requires a deep understanding of quantum mechanics and physics procedures. This work presents a generic quantum “black box” for entropy calculation. It does not depend on the data type and can be applied to building and maintaining machine learning models. The method has two main advantages. First, it is accessible to those without preliminary knowledge of quantum computing. Second, it is based on the quantum circuit with a constant depth of three, which is equivalent to three operations the circuit would perform to achieve the same result. We implemented our method using the IBM simulator and tested it over different types of input. The results showed a high correspondence between the classical and quantum computations that raised an error of up to $8.8e-16$ for different lengths and types of information.

Keywords Entropy · Quantum computing · Information retrieval

1 Introduction and related work

Quantum computing (QC) is one of the most promising fields in computation and has taken an important place in international research (Ying 2010). QC is based on the physics theorem, which assumes that an electron can behave simultaneously as a wave and a particle (Robertson 1943). However, there are some difficulties in building and maintaining the superposition of quantum computers due to the sensitivity of the computers to noise and decoherence (Bennett et al. 1997; De Wolf 2019). Some technology companies have quantum computers and have invested in developing this field (Zeng et al. 2017). Throughout the years, arguments on the advantages and disadvantages of QC have been raised and are still discussed today (Boyer et al. 1998; De Wolf 2019).

A few QC algorithms have been developed throughout the years to answer different problems. Grover’s algorithm is one example, which was developed to solve the problem of finding a value in an unsorted array (Lavor et al. 2003; Leunenberger and Loss 2003). An additional significant algorithm is Shor’s algorithm. Shor’s algorithm was discovered in 1994 and proposes a solution for integer factorization. Shor used the advantages of QC to solve a problem in polynomial time, rather than exponential time in a classical computer (Hayward 2008). Theoretically, Shor’s algorithm breaks public-key cryptography schemes like the widely used Rivest–Shamir–Adleman (RSA) scheme. The RSA is a public-key cryptosystem used for secure data transmission and is based on the assumption that factoring large integers is computationally intractable (Milanov 2009). Therefore, it may be feasible to defeat RSA by constructing and maintaining a large quantum computer (Thombre and Jajodia 2021).

As mentioned above, quantum computers significantly decrease computing complexity. Therefore, they can perform wider operations than classical computers in parallel processes (Biamonte et al. 2017; Wiebe 2020). However, algorithms can use a combination of classical computers and quantum computers and are not exclusive to one type of computer or the other (Buffoni and Caruso 2021). Thus, the combination of QC and classical computing yields a young but rapidly growing field, quantum machine learning (QML). Generally,

✉ Michal Koren
michal.koren@shenkar.ac.il
Oded Koren
odedkoren@shenkar.ac.il
Or Peretz
or.peretz@shenkar.ac.il

¹ School of Industrial Engineering and Management,
Shenkar—Engineering. Design. Art, Anne Frank,
12 Ramat Gan, Israel

transforming a classical machine learning algorithm into QC requires implementing the logic of the classical machine learning algorithm with circuits composed of quantum gates (Benedetti et al. 2019; Alchieri et al. 2021). Recently, studies have presented quantum algorithms for learning random variables (González et al. 2022; Pirhooshayan and Terlaky 2021), building a quantum convolutional network to learn images (Hur et al. 2022; Tüysüz et al. 2021), developing generative adversarial networks (GANs) and transfer learning (Assouel et al. 2022; Azevedo et al. 2022; Zoufal et al. 2021), and reinforcement learning implementation (Dalla et al. 2022).

In physics, entropy is essential for describing uncertainty in the state of matter (Bein 2006). In recent years, the concept of entropy has become increasingly important in the theory of information with the development of information technology. Thus, information can be quantified by measuring the amount of data in events, random variables, and distributions (Wehrl 1978). At the same time, probabilities are used to quantify information, which is why information theory is related to probability theory. Furthermore, information measurements are widely used in artificial intelligence and machine learning, such as in constructing decision trees and optimizing classifier models (Kapur and Kesavan 1992). As such, there is a significant relationship between information theory and machine learning, and a practitioner must be familiar with some of the basic concepts from the field (Huang et al. 2022; Liu et al. 2022). Relatedly, in data mining and machine learning, entropy represents a model's degree of unpredictability or impurity. Therefore, if it is easier to draw a valuable conclusion from a piece of information, then the entropy will be lower. On the other hand, if the entropy is higher, it will be more challenging to make conclusions based on that information (Kaufmann et al. 2020; Kaufmann and Vecchio et al. 2020).

This work presents a quantum “black box” for entropy calculation. It is a generic procedure, regardless of data type, and can be applied for information analysis, ML algorithms, and more. Section 2 describes the procedure's correctness and general implementation using quantum logic circuits. The central innovative aspect of this method is to allow users without any background in QC to utilize the capabilities of a quantum computer for their specific needs without the need to build the quantum circuits and transform the problem from classical to quantum computation. Therefore, this “black box” is accessible to those without preliminary knowledge of QC. Moreover, our quantum “black box” has a fixed depth of three, equivalent to the number of steps done by the quantum computer that runs the circuit. Comparing it to classical computer computation, it does not depend on the input size as it is based on amplitude encoding, which encodes the input as a single state of the quantum circuit. Section 3 presents a

case study that compares our method to classical computer results. Section 4 describes the main conclusions and suggestions for future research.

2 Quantum entropy “black box”

This section presents and describes a new method for quantum entropy calculation. It is aimed at making QC accessible and enables entropy calculation using quantum computers. The method assumes that the state vector (in a single time phase) includes the probability that the circuit will end in a specific state. First, we will describe the method and its procedure. Then, we will present the implementation and correctness of the method.

2.1 Quantum logic and gates

Let $v = (v_1, v_2, \dots, v_n)$ be the input vector that represents the occurrences of each item (i.e., each $v_i \in \mathbb{N} \cup \{0\}$ represents the number of occurrences of the i th item). To begin, the algorithm transforms v to an amplitude encoding by concatenating all n items into a single amplitude vector. Let \tilde{v} be the amplitude vector, such that $|\tilde{v}|^2 = 1$. The normalization constant, denoted as \tilde{A} , satisfies

$$\tilde{A} = \frac{1}{\sqrt{\sum_{i=1}^n v_i}}$$

The input vector can be represented in the computational basis as $\tilde{v} = \sum_{i=1}^n \sqrt{v_i} |i\rangle$. Since a quantum system of n qubits provides 2^n amplitudes, encoding \tilde{v} requires the use of $\lceil \log_2 n \rceil + 1$ qubits. It is important to note that in cases where the length of \tilde{v} is not to the power of two, zeros were added as their values do not change the entropy calculation.

Next, the algorithm creates a quantum circuit using $\lceil \log_2 n \rceil + 1$ qubits, initializes the states by \tilde{v} elements, and applies the unitary gate $U\left(\frac{\pi}{2}, 0, \pi\right)$ on each qubit (equivalent to the Hadamard gate) to transform it into superposition. The vector is presented as an amplitude encoding; thus, each state holds the probability of the input item. Let $|\psi\rangle$ be the state vector achieved in this time phase. Thus, $|\psi\rangle$ is a vector of size n and represents the complex form probabilities of the original vector items. Next, the algorithm creates PE, a parameterized vector of size n (i.e., a vector that defines parameters according to the values assigned while running the quantum circuit), such that the items of PE are the coefficients of $H|\psi\rangle$ multiplied by $\log_2(e) \approx 1.4427$. As PE is a parameterized vector, we applied a logarithm rotation and achieved the following:

$$PE = \log_2(e) \left[\log_e \left(\frac{\sqrt{v_i}}{\sqrt{\sum_{i=1}^n v_i}} \right) \right]$$

Note that PE cannot be performed as stand-alone operation since it does not represent a state vector. For that, let W be a square and invertible diagonal matrix of size n , where W_{ii} holds $|\psi\rangle$ coefficients, and $W_{ij}=0$ for each $i \neq j$. The method applies W on PE to calculate the multiplication of the state probabilities in their logarithm rotation vector. At the end of the quantum circuit, the method applies the unitary gate $U\left(\frac{\pi}{2}, 0, \pi\right)$ on each qubit and returns the output vector. Last, the method uses classical computer computation to calculate the vector sum (i.e., the total entropy of \tilde{v}).

Notes

1. The PE and W gates are described earlier in this section. The proof of its correctness is detailed in Section 2.2. Note that PE and W are circled (in Fig. 1) and defined as an operation to satisfy the invertibility conditions.
2. The dashed lines describe the entry and exit of the qubits from the superposition.
3. We used the IBM simulators (with the Qiskit Library for Python; Cross 2018) to avoid noise and to be able to sample the state vector in each time phase in the circuit. This differs from quantum computers as each observation/measurement causes collapse of the quantum circuit.
4. Fig. 1 describes the quantum circuit over three qubits, although generalization to a higher dimension can be done with tensor products.

QunatumEntropyCircuit(\tilde{v})

- Allocate $\lfloor \log_2 n \rfloor + 1$ qubits
- $|\psi\rangle \leftarrow$ set \tilde{v} as initial state
- $|\psi\rangle \leftarrow H|\psi\rangle$
- Apply $PE(H|\psi\rangle)$ the parametric gate
- Apply logarithm rotation on $PE(H|\psi\rangle)$ and multiply in $\log_2(e)$
- Apply W gate on the current state
- Return the current state

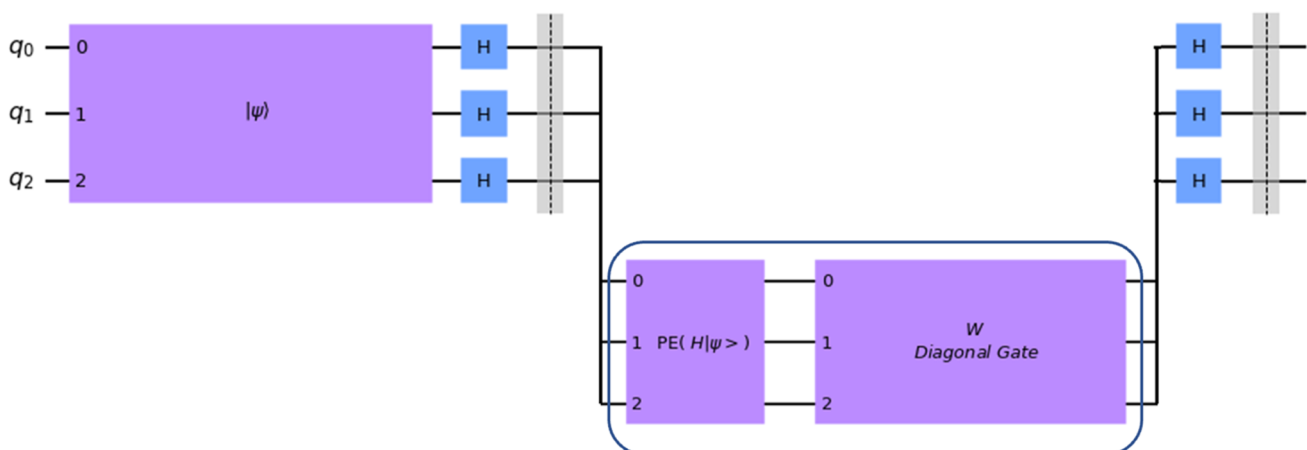


Fig. 1 The quantum circuit of entropy calculation

Entropy(v)

- $\tilde{v} \leftarrow \emptyset$
- **For** $v_i \in v$
 - $\tilde{v} \leftarrow \tilde{v} \cup \left\{ \frac{\sqrt{v_i}}{\sqrt{\sum_{i=1}^n v_i}} \right\}$
- $R \leftarrow \text{QunatumEntropyCircuit}(\tilde{v})$
- **Return** $\sum_{r \in R} |r|$

2.2 Correctness

Let $v = (v_1, v_2, \dots, v_n)$ be the input vector that represents the occurrences of each item (i.e., each $v_i \in \mathbb{N} \cup \{0\}$ represents the number of occurrences of the i th item). Let \tilde{A} be the normalization constant. Applying the square root of each item in v , we get

$$\left(\sum_{i=1}^n (\sqrt{v_i})^2 \right) \tilde{A}^2 = 1$$

$$\tilde{A}^2 = \frac{1}{\sum_{i=1}^n v_i} \tilde{A} = \frac{1}{\sqrt{\sum_{i=1}^n v_i}}$$

The method transforms v to an amplitude vector, denoted as \tilde{v} , such that each $v_i \in v$ is converted to $\frac{\sqrt{v_i}}{\sqrt{\sum_{i=1}^n v_i}}$. Therefore, it satisfies the following:

$$\tilde{v} = \tilde{A} \cdot \sum_{i=1}^n \sqrt{v_i}$$

$$|\tilde{v}|^2 = \sum_{i=1}^n \left(\frac{\sqrt{v_i}}{\sqrt{\sum_{i=1}^n v_i}} \right)^2 = \sum_{i=1}^n \frac{v_i}{\sum_{i=1}^n v_i} = \frac{\sum_{i=1}^n v_i}{\sum_{i=1}^n v_i} = 1$$

Let $|\psi\rangle$ be the initialized state vector. The method set \tilde{v} to the initial states and applies the U gate with the parameters $\theta = \frac{\pi}{2}, \phi = 0, \lambda = \pi$, which is equivalent to applying the Hadamard gate to move the states into superposition. Thus, the current quantum circuit is $H|\psi\rangle$. Since the normalization constant sums up to one, the coefficients of $H|\psi\rangle$ can describe the probability of each state and have the form $\sqrt{p_i} |i\rangle$, where p_i is the probability of the i th item in the computational basis.

Let PE be a parametric vector of size n (i.e., a vector that defines parameters according to their values assigned while running the quantum circuit), such that each PE _{i} item is the item of $H|\psi\rangle$ multiplied by $\log_2(e) \approx 1.4427$. Thus,

$$\text{PE} = \log_2(e) \cdot H | \psi \rangle$$

In quantum computing, the logarithms are in a natural base (i.e., e) since all basic computations can be described by polar coordinates. To multiply PE in $\log_2(e)$, we converted the calculations to base two (i.e., the binary base):

$$\log_e(a) = \frac{\log_2(a)}{\log_2(e)} \iff \log_2(a) = \log_2(e) \cdot \log_e(a)$$

Therefore, PE is a vector that represents the logarithm of the state coefficients (i.e., the probabilities):

$$\text{PE} = \log_2(e) \left[\log_e \left(\frac{\sqrt{v_i}}{\sqrt{\sum_{i=1}^n v_i}} \right) \right]$$

Let W be a diagonal gate of size n , including all p_i elements (i.e., the square coefficients of $|\psi\rangle$):

$$W = \begin{pmatrix} \frac{v_1}{\sum_{i=1}^n v_i} & 0 & \dots & 0 \\ \vdots & \frac{v_2}{\sum_{i=1}^n v_i} & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \frac{v_n}{\sum_{i=1}^n v_i} \end{pmatrix}$$

Applying $W \cdot (\text{PE})$ returns a vector of size n , in which each element represents the multiplication of the W diagonal in the logarithm parametric vector. Last, applying H again yields

$$W \cdot (\text{PE}) \cdot H = W \cdot \log_2(e) \cdot H | \psi \rangle H$$

Given that $HH = I$, we get an output of

$$W \cdot \log_2(e) \cdot H | \psi \rangle H =$$

$$= W \cdot \log_2(e) \cdot | \psi \rangle =$$

$$= \log_2(e) \cdot W \cdot | \psi \rangle$$

where $W \cdot | \psi \rangle$ is the multiplication of the state probabilities in the current state (i.e., the logarithm rotation) and $\log_2(e)$ normalized the result into base two.

3 Case study

This section presents the case study and experiments of the entropy calculation using our method compared to classical computer computation. Each experiment was simulated using an IBM simulator with 1024 shots. In cases based on the state vector of the quantum circuit, we used the state

achieved by most of the shots. For the demonstration's simplification, we first describe a simple use case of entropy calculation of a numerical vector and detail each state and operation in the quantum circuit. Then, we present an entropy calculation of a given text.

3.1 Simple occurrences vector

Let $v = [4, 3, 1, 6]$ be the vector of occurrences of size four, such that the first item appeared four times, the second item appeared three times, and so on. The classical computer computation for entropy yielded

$$-\frac{2}{7}\log_2\left(\frac{2}{7}\right) - \frac{3}{14}\log_2\left(\frac{3}{14}\right) - \frac{1}{14}\log_2\left(\frac{1}{14}\right) - \frac{3}{7}\log_2\left(\frac{3}{7}\right) = 1.788$$

The quantum circuit converted v into an amplitude vector \tilde{v} , such that each $v_i \in v$ was assigned to $\frac{\sqrt{v_i}}{\sqrt{\sum_{i=1}^n v_i}}$. Applying the Hadamard gate and pushing \tilde{v} into the superposition yielded a state vector $|\psi\rangle$ of

$$|\psi\rangle = \frac{\sqrt{14}}{7} |00\rangle + \frac{\sqrt{42}}{14} |01\rangle + \frac{\sqrt{14}}{14} |10\rangle + \frac{\sqrt{21}}{7} |11\rangle$$

Next, the method created PE, the parameterize vector equal to the logarithm rotation of $|\psi\rangle$ multiplied in $\log_2(e)$:

$$\text{PE}(|\psi\rangle) = \begin{pmatrix} -1.807i \\ -2.222i \\ -3.807i \\ -1.222i \end{pmatrix}$$

Since $\text{PE}(|\psi\rangle)$ is a parameterized vector, we multiplied it by the diagonal matrix W to ensure an invertible state operation:

$$W = \begin{pmatrix} 0.285 & 0 & 0 & 0 \\ 0 & 0.215 & 0 & 0 \\ 0 & 0 & 0.072 & 0 \\ 0 & 0 & 0 & 0.428 \end{pmatrix}$$

$$\begin{pmatrix} 0.285 & 0 & 0 & 0 \\ 0 & 0.215 & 0 & 0 \\ 0 & 0 & 0.072 & 0 \\ 0 & 0 & 0 & 0.428 \end{pmatrix} \cdot \begin{pmatrix} -1.807 \\ -2.222 \\ -3.807 \\ -1.222 \end{pmatrix} = \begin{pmatrix} -0.285 \cdot 1.807i \\ -0.215 \cdot 2.222i \\ -0.072 \cdot 3.807i \\ -0.428 \cdot 1.222i \end{pmatrix} = \begin{pmatrix} -0.515i \\ -0.477i \\ -0.275i \\ -0.523i \end{pmatrix}$$

Last, the classical computer computed the absolute sum value of the matrix, which represented the total entropy observed:

$$0.515 + 0.477 + 0.275 + 0.523 = 1.788$$

The error (i.e., difference) between both outputs is $3.2e-21$, indicating a high level of agreement between the classic and quantum computations.

3.2 Entropy of text

For the demonstration of our method on text, we used the information presented in Section 1 of this work (i.e., "Introduction and related work"). We preprocessed the text and converted it into an array of occurrences, where the first item was the number of "a" occurrences, the second item was "b", and so on. As an input vector, we received a vector of size 24 with an entropy of 4.155, calculated by classical computer computation.

First, for the quantum circuit, we added eight zero values to ensure the input had a size that was to the power of two (i.e., 32) and then applied the Hadamard gate and pushed it into superposition. It yielded a state vector $|\psi\rangle$ of size 32; hence, $|\psi\rangle$ coefficients represented the probabilities of each input element. Next, the quantum circuit used the parametric vector and the diagonal gate to perform a logarithm rotation of the input vector. In this case, the output of the quantum circuit was a vector of size 32, representing all sub-multiplications of probability in its logarithm. The quantum calculation output was similar to the classical computer computation and presented an entropy of 4.155 after 1024 shots.

To understand the level of agreement between the two methods, we examined the differences between the computed values. The error (i.e., the difference) was $8.8e-16$, which is relatively low for the input of such a long text. The significant difference was that the quantum circuit performed three operations to achieve the desired entropy, which was faster than the classical computer computation.

4 Analysis

This section provides a comparison and analysis between the proposed method and other existing methods for

calculating entropy. First, we describe the methods we used for the comparison. Then, we present the results of our method over four types of input.

4.1 Entropy calculation methods

We examined the following methods to demonstrate the results:

1. Shannon entropy (Shannon 1948)—given a random variable X , which takes values of $\{x_1, x_2, \dots, x_m\}$ under sample space Ω , the Shannon entropy, denoted as $H_S(X)$, is defined by

$$H_S(X) = - \sum_{x_i \in X} p(x_i) \cdot \log_2(p(x_i))$$

2. von Neumann entropy (Von Neumann 1955; Nielsen and Chuang 2010)—let ρ be the density matrix of a quantum state. The following are two equivalent approaches to calculating the von Neumann entropy, denoted as $S(\rho)$:

- a. Assuming that $\|\rho - I\| < 1$, where I is the identity matrix, then the following power series is convergent and defines the logarithm of ρ :

$$\log(\rho) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(\rho - I)^k}{k} S(\rho) = -\text{Tr}(\rho \cdot \log_2(\rho))$$

- b. Let $\{\lambda_1, \dots, \lambda_k\}$ be the set of eigenvalues of the matrix ρ . Then, the von Neumann entropy can also be defined as

$$S(\rho) = - \sum_{i=1}^k \lambda_i \cdot \log_2(\lambda_i)$$

To demonstrate the equivalence of both approaches, let ρ be the density matrix of the quantum state, such as

$$\rho = \frac{1}{3} |1\rangle\langle 1| + \frac{2}{3} |0\rangle\langle 0| = \begin{pmatrix} \frac{5}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} \end{pmatrix}$$

On the one hand, the eigenvalues of ρ are $\lambda_1 = 0.872$, $\lambda_2 = 0.127$. By applying the second approach to calculate the von Neumann entropy, the following is obtained:

$$S(\rho) = - \sum_{i=1}^k \lambda_i \cdot \log_2(\lambda_i) = 0.55$$

On the other hand, a logarithm base conversion must be used to acquire the appropriate base:

$$\log_2(\rho) = \frac{\log_e(\rho)}{\log_e(2)} = \frac{\begin{pmatrix} -0.237 & 0.430 \\ 0.430 & -1.959 \end{pmatrix}}{\log_e(2)} = \begin{pmatrix} -0.343 & 0.620 \\ 0.620 & -2.826 \end{pmatrix}$$

Table 1 Comparison of the proposed method to the Shannon entropy and von Neumann Entropy

	Quantum entropy	Shannon entropy	von Neumann entropy
Input A	1.788	1.788	1.779
Input B	4.155	4.156	4.423
Input C	4.680	4.679	4.669
Input D	3.313	3.318	3.331

Thus, the von Neumann entropy is

$$S(\rho) = -\text{Tr}(\rho \cdot \log_2(\rho)) = -\text{Tr}\left(\begin{pmatrix} -0.182 & 0.046 \\ 0.046 & -0.367 \end{pmatrix}\right) = 0.55$$

4.2 Results

For the analysis of our method, Table 1 presents the comparison to other existing methods over the following inputs:

1. Input A—a simple occurrence vector, as described in Section 3.1
2. Input B—a text, as described in Section 3.2
3. Input C—a randomized text of size 5000 consisting of uppercase and lowercase letters
4. Input D—a randomized text of size 1000 consisting of digits only

As demonstrated in Table 1, all the entropy calculation methods obtained results in an (almost) identical range without extreme anomalies or noises. The differences between our method and the Shannon entropy were minor. Therefore, it can be concluded that there is an agreement between them. When comparing these results to the von Neumann entropy, slightly more significant deviations are obtained (e.g., a difference of 0.268 on input B). This difference might be due to the different calculation methods or the noises created throughout the 1024 simulations in the quantum algorithm.

5 Conclusions and discussion

This study proposes a novel quantum “black box” for entropy calculation. The presented procedure is generic and can be applied to information analyses, machine learning algorithms, and more. The method involves amplitude encoding, a key component of quantum computing, representing a vector’s probability. Its main innovation is the use of quantum computers to calculate entropy as a “black box” without having to build quantum circuits or transform the problem from classical to quantum computation. As a result, this “black box” is accessible to those without a previous understanding of quantum computing. The following are the main conclusions:

1. Our method calculated the entropy of different data types with the same precision as a classic computer. The significant difference was the amplitude encoding, which represents the dataset as a complete vector that yields the probabilities of the input. Our method used amplitude encoding for entropy estimation, although this can be generalized to any measure based on stochastic elements.
2. Our quantum “black box” has a fixed depth of three, equivalent to the number of steps performed by the quantum computer that runs the circuit. Circuit depth matters because qubits have finite coherence time. Depth complexity is not independent of gate complexity because a circuit with many gates is also likely to have considerable depth. Thus, circuit depth can increase due to both the algorithm’s structure and the physical limitations of the hardware. When comparing it to classical computer computation, our method does not depend on the input size since it is based on amplitude encoding, which encodes the input as a single state of the quantum circuit.

This study presented two main issues that must be addressed in future studies. First, we tested our method using the IBM simulator to avoid noise and to be able to sample a state vector without collapsing the circuit. Since this study was designed to find a method for entropy calculation in quantum computing, future studies should examine the implementation and evaluation of our method using a quantum computer. Second, besides entropy, many “black boxes” can help build and maintain learning algorithms, such as information gain, weighted and conditional entropy, and distance metrics. Future studies should focus on developing those “boxes” to make quantum computing more accessible.

Data availability The data used in this study is available and accessible in UCI Machine Learning Repository–Datasets (<https://archive.ics.uci.edu/ml/datasets.php>).

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alchieri L, Badalotti D, Bonardi P, Bianco S (2021) An introduction to quantum machine learning: from quantum logic to quantum deep learning. *Quantum Mach Intell* 3:28. <https://doi.org/10.1007/s42484-021-00056-8>
- Assouel A, Jacquier A, Kondratyev A (2022) A quantum generative adversarial network for distributions. *Quantum Mach Intell* 4:28. <https://doi.org/10.1007/s42484-022-00083-z>
- Azevedo V, Silva C, Dutra I (2022) Quantum transfer learning for breast cancer detection. *Quantum Mach Intell* 4:5. <https://doi.org/10.1007/s42484-022-00062-4>
- Bein B (2006) Entropy. *Best Pract Res Clin Anaesthesiol* 20:101–109. <https://doi.org/10.1016/j.bpa.2005.07.009>
- Benedetti M, Lloyd E, Sack S, Fiorentini M (2019) Parameterized quantum circuits as machine learning models. *Quantum Sci Technol* 4:043001. <https://doi.org/10.1088/2058-9565/ab4eb5>
- Bennett CH, Bernstein E, Brassard G, Vazirani U (1997) Strengths and weaknesses of quantum computing. *SIAM J Comput* 26:1510–1523. <https://doi.org/10.1137/S0097539796300933>
- Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S (2017) Quantum machine learning. *Nature* 549:195–202. <https://doi.org/10.1038/nature23474>
- Boyer M, Brassard G, Høyer P, Tapp A (1998) Tight bounds on quantum searching. *Fortschritte der Phys* 46:493–505. [https://doi.org/10.1002/\(SICI\)1521-3978\(199806\)46:4/5<493::AID-PROP493>3.0.CO;2-P](https://doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P)
- Buffoni L, Caruso F (2021) New trends in quantum machine learning (a). *Europhys Lett* 132:60004. <https://doi.org/10.1209/0295-5075/132/60004>
- Cross A (2018) The IBM Q experience and QISKit open-source quantum computing software. *APS March Meet Abstr* 2018:L58–L003
- Dalla Pozza N, Buffoni L, Martina S, Caruso F (2022) Quantum reinforcement learning: the maze problem. *Quantum Mach Intell* 4:11. <https://doi.org/10.1007/s42484-022-00068-y>
- De Wolf R (2019) Quantum computing: lecture notes. arXiv preprint arXiv:1907.09415. <https://arxiv.org/abs/1907.09415>
- González FA, Gallego A, Toledo-Cortés S, Vargas-Calderón V (2022) Learning with density matrices and random features. *Quantum Mach Intell* 4:23. <https://doi.org/10.1007/s42484-022-00079-9>
- Hayward M (2008) Quantum computing and Shor’s algorithm. Macquarie University Mathematics Department, Sydney
- Huang EW, Lee WJ, Singh SS, Kumar P, Lee CY, Lam TN, Chin HH, Lin BH, Liaw PK (2022) Machine-learning and high-throughput

- studies for high-entropy materials. *Mater Sci Eng R Rep* 147:100645. <https://doi.org/10.1016/j.mser.2021.100645>
- Hur T, Kim L, Park DK (2022) Quantum convolutional neural network for classical data classification. *Quantum Mach Intell* 4:3. <https://doi.org/10.1007/s42484-021-00061-x>
- Kapur JN, Kesavan HK (1992) Entropy optimization principles and their applications. In: Singh VP, Fiorentino M (eds) *Entropy and energy dissipation in water resources*. Springer, Dordrecht, pp 3–20
- Kaufmann K, Maryanovsky D, Mellor WM, Zhu C, Rosengarten AS, Harrington TJ, Oses C, Toher C, Curtarolo S, Vecchio KS (2020) Discovery of high-entropy ceramics via machine learning. *NPJ Comput Mater* 6:42. <https://doi.org/10.1038/s41524-020-0317-6>
- Kaufmann K, Vecchio KS (2020) Searching for high entropy alloys: a machine learning approach. *Acta Mater* 198:178–222. <https://doi.org/10.1016/j.actamat.2020.07.065>
- Lavor C, Manssur L, Portugal R (2003) Grover's algorithm: quantum database search. arXiv preprint quant-ph/0301079. <https://arxiv.org/pdf/quant-ph/0301079.pdf>
- Leuenberger MN, Loss D (2003) Grover algorithm for large nuclear spins in semiconductors. *Phys Rev B* 68:165317. <https://doi.org/10.1103/PhysRevB.68.165317>
- Liu X, Zhang J, Pei Z (2022) Machine learning for high-entropy alloys: progress, challenges and opportunities. *Prog Mater Sci* 131:101018. <https://doi.org/10.1016/j.pmatsci.2022.101018>
- Milanov E (2009) The RSA algorithm. RSA Laboratories https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf
- Nielsen MA, Chuang IL (2010) *Quantum computation and quantum information*. Cambridge University Press, Cambridge
- Pirhooshyaran M, Terlaky T (2021) Quantum circuit design search. *Quantum Mach Intell* 3:25. <https://doi.org/10.1007/s42484-021-00051-z>
- Robertson JK (1943) The role of physical optics in research. *Am J Phys* 11:264–271. <https://doi.org/10.1119/1.1990496>
- Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27:379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- Thombre R, Jajodia B (2021) Experimental analysis of attacks on RSA & Rabin cryptosystems using quantum Shor's algorithm. *AIJR Proc*:587, 596. <https://doi.org/10.21467/proceedings.114.74>
- Tüysüz C, Rieger C, Novotny K, Demirköz B, Dobos D, Potamianos K, Vallecorsa S, Vilmant JR, Forster R (2021) Hybrid quantum classical graph neural networks for particle track reconstruction. *Quantum Mach Intell* 3:29. <https://doi.org/10.1007/s42484-021-00055-9>
- Von Neumann J (1955) *Mathematical foundations of quantum mechanics* (RT Beyer, Trans.; 1st ed.). Princeton University Press Princeton (Original work published 1932)
- Wehrl A (1978) General properties of entropy. *Rev Mod Phys* 50(2):221–260. <https://doi.org/10.1103/RevModPhys.50.221>
- Wiebe N (2020) Key questions for the quantum machine learner to ask themselves. *New J Phys* 22:091001. <https://doi.org/10.1088/1367-2630/abac39>
- Ying M (2010) Quantum computation, quantum theory and AI. *Artif Intell* 174:162–176. <https://doi.org/10.1016/j.artint.2009.11.009>
- Zeng W, Johnson B, Smith R, Rubin N, Reagor M, Ryan C, Rigetti C (2017) First quantum computers need smart software. *Nature* 549:149–151. <https://doi.org/10.1038/549149a>
- Zoufal C, Lucchi A, Woerner S (2021) Variational quantum Boltzmann machines. *Quantum Mach Intell* 3:7. <https://doi.org/10.1007/s42484-020-00033-7>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.