

Hybrid quantum-classical multi-agent decision-making framework based on hierarchical Bayesian networks in the noisy intermediate-scale quantum era

Hao Shi(石皓), Chenghao Han(韩成豪), Peng Wang(王鹏), and Ming Zhang(张明)
Citation: Chin. Phys. B, 2025, 34 (**12**): 120304. DOI: 10.1088/1674-1056/adefd7
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Information scrambling in a partially confined quantum link model

Yifan Luo(罗祎帆), Zheng Tang(唐正), Li Chen(陈立), and Wei Zheng(郑炜)
Chin. Phys. B, 2025, 34 (**11**): 110301. DOI: 10.1088/1674-1056/ae00b1

Progressive quantum algorithm for maximum independent set with quantum alternating operator ansatz

Xiao-Hui Ni(倪晓慧), Ling-Xiao Li(李凌霄), Yan-Qi Song(宋燕琪), Zheng-Ping Jin(金正平), Su-Juan Qin(秦素娟), and Fei Gao(高飞)
Chin. Phys. B, 2025, 34 (**7**): 070304. DOI: 10.1088/1674-1056/addd83

Domain adaptation method inspired by quantum convolutional neural network

Chunhui Wu(武春辉), Junhao Pei(裴骏豪), Yihua Wu(吴逸华), Anqi Zhang(张安琪), and Shengmei Zhao(赵生妹)
Chin. Phys. B, 2025, 34 (**7**): 070302. DOI: 10.1088/1674-1056/adc7ed

Distributed quantum circuit partitioning and optimization based on combined spectral clustering and search tree strategies

Zilu Chen(陈子禄), Zhijin Guan(管致锦), Shuxian Zhao(赵书娴), and Xueyun Cheng(程学云)
Chin. Phys. B, 2025, 34 (**5**): 050305. DOI: 10.1088/1674-1056/adbada

Effect of pseudo-random number on the security of quantum key distribution protocol

Xiao-Liang Yang(杨晓亮), Yu-Qing Li(李毓擎), and Hong-Wei Li(李宏伟)
Chin. Phys. B, 2025, 34 (**2**): 020301. DOI: 10.1088/1674-1056/ad99c9

Hybrid quantum–classical multi-agent decision-making framework based on hierarchical Bayesian networks in the noisy intermediate-scale quantum era

Hao Shi(石皓)[1], Chenghao Han(韩成豪)[1], Peng Wang(王鹏)[2], and Ming Zhang(张明)[1,†]

[1] *College of Intelligence Science and Technology, National University of Defense Technology, Changsha* 410073, *China*
[2] *College of Science, National University of Defense Technology, Changsha* 410073, *China*

(Received 9 April 2025; revised manuscript received 26 June 2025; accepted manuscript online 15 July 2025)

Although quantum Bayesian networks provide a promising paradigm for multi-agent decision-making, their practical application faces two challenges in the noisy intermediate-scale quantum (NISQ) era. Limited qubit resources restrict direct application to large-scale inference tasks. Additionally, no quantum methods are currently available for multi-agent collaborative decision-making. To address these, we propose a hybrid quantum–classical multi-agent decision-making framework based on hierarchical Bayesian networks, comprising two novel methods. The first one is a hybrid quantum–classical inference method based on hierarchical Bayesian networks. It decomposes large-scale hierarchical Bayesian networks into modular subnetworks. The inference for each subnetwork can be performed on NISQ devices, and the intermediate results are converted into classical messages for cross-layer transmission. The second one is a multi-agent decision-making method using the variational quantum eigensolver (VQE) in the influence diagram. This method models the collaborative decision-making with the influence diagram and encodes the expected utility of diverse actions into a Hamiltonian and subsequently determines the intra-group optimal action efficiently. Experimental validation on the IonQ quantum simulator demonstrates that the hierarchical method outperforms the non-hierarchical method at the functional inference level, and the VQE method can obtain the optimal strategy exactly at the collaborative decision-making level. Our research not only extends the application of quantum computing to multi-agent decision-making but also provides a practical solution for the NISQ era.

Keywords: quantum Bayesian networks, multi-agent decision-making, hybrid quantum–classical algorithms, hierarchical Bayesian networks

PACS: 03.67.Ac, 03.67.–a

DOI: 10.1088/1674-1056/adefd7 **CSTR:** 32038.14.CPB.adefd7

1. Introduction

In the multi-agent system, agents' autonomous decision-making capabilities are vital in the dynamic and uncertain environments. Each agent not only possesses independent perception and decision-making capabilities but also shares information with other agents and engages in collaborative decision-making.

In the field of autonomous decision-making, prevalent methodologies include neural networks,[1] fuzzy reasoning,[2] rough set theory[3] and Bayesian networks.[4] Among these, Bayesian networks are particularly favored for their strong performance. As a probabilistic graphical model,[5] Bayesian networks reveal data relationships through their graphical structure and offer two significant advantages. (1) They enable modular decomposition of complex tasks into distributed subtasks, thereby reducing computational complexity.[5] (2) They provide a universal framework that adapts to diverse scenarios and excels at uncertainty inference for complex systems.[6] Furthermore, the inference process of Bayesian networks shares similarities with the reasoning mechanism of the human brain,[6] making it an ideal choice for agent decision-making. In Bayesian network inference algorithms, there are exact methods and approximate methods.[5] Exact methods are suitable for small-scale networks and offer high computational accuracy. In contrast, approximate methods can address large-scale network inference, yet they have relatively low computational accuracy. When the multi-agent systems encounter high-dimensional state spaces, complex interaction relationships, and real-time requirements, traditional methods often face multiple challenges, including high computational complexity and slow decision-making speed.

Quantum computing provides a new computation paradigm by leveraging quantum mechanical principles such as quantum superposition and quantum entanglement.[8] Algorithms such as Shor's algorithm,[9] Grover's algorithm,[10] and the HHL algorithm[11] have demonstrated the acceleration advantages in solving certain problems. However, the lack of fault-tolerant quantum computers makes it difficult to apply these algorithms to practical tasks. In the noisy intermediate-scale quantum (NISQ) era,[12] due to the limited

†Corresponding author. E-mail: zhangming@nudt.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

number of qubits and persistent quantum noise on quantum hardware, the hybrid quantum–classical computing architecture has been proposed. The most representative examples are variational quantum algorithms,[13–15] which have successfully solved combinatorial optimization problems through the collaboration between parameterized shallow quantum circuits and classical optimizers.

Therefore, based on the hybrid quantum–classical computing architecture, exploring how to utilize limited quantum computing resources to perform decision-making tasks in multi-agent systems is a promising research area. The tools and methods involved include quantum Bayesian networks,[16,17] quantum machine learning,[18] quantum reinforcement learning,[19] quantum optimization algorithms,[13] quantum games,[20] etc. Among them, quantum Bayesian networks serve as a fundamental tool for handling uncertain inference.

Many scholars have constructed quantum Bayesian networks to solve inference and decision-making problems. The concept of quantum Bayesian networks was initially proposed by Tucci.[16] As a quantum version of classical Bayesian networks, Tucci proposed using the amplitudes of quantum states to express conditional probabilities. Later, Ozols *et al.* proposed quantum rejection sampling using the principle of amplitude amplification.[21] Low *et al.* implemented the quantum inference on Bayesian networks, applying the amplitude amplification technique to achieve a quadratic acceleration compared to classical approximate inference. Wiebe *et al.* proposed a new form of adaptive approximate quantum Bayesian inference to solve the problem of updating probability distributions, which has a polynomial acceleration compared to classical inference.[17] Moreira *et al.* proposed a quantum-like Bayesian network that can better explain and handle scenarios with high uncertainty and address some situations that violate classical probability theory and logic. It has clear advantages over classical algorithms in certain tasks, but it still remains an NP-hard problem.[22,23] Borujeni *et al.* proposed a general quantum circuit capable of expressing Bayesian networks.[24] Nguyen *et al.* introduced a class of quantum Bayesian neural networks, which have the desirable properties of Bayesian inference and also have prediction performance comparable to classical algorithms.[25] Some researchers have utilized quantum Bayesian networks to solve practical problems. Harikrishnakumar *et al.* used a 4-node quantum Bayesian network to predict the demand for shared bicycles on weekdays and weekends.[26] Carrascal *et al.* analyzed the failure risks in the distribution network using a quantum Bayesian network.[27] Fathallah *et al.* addressed the challenges of limited quantum hardware resources and noise interference in the quantum Bayesian network by proposing an optimized representation method of quantum circuits to reduce the circuit width.[28] Nayak *et al.* introduced a hybrid quantum approximate inference technique that outperforms quantum inference and classical inference in diagnostic and rare event queries.[29] Oliveira *et al.* extended Low's algorithm to the decision aspect, introduced a utility function, and proposed a quantum Bayesian decision method that also achieved quadratic acceleration.[30]

Variational quantum algorithms (VQAs) are a class of hybrid quantum–classical algorithms that can solve combinatorial optimization problems on NISQ devices, which require fewer qubits and shallower quantum circuits.[31] The core idea is to use parameterized quantum circuits to prepare quantum states and tune these parameters through classical optimizers to minimize the expectation value of a Hamiltonian. The quantum approximate optimization algorithm (QAOA)[15] and the variational quantum eigensolver (VQE)[32] are two well-known variants of VQAs. The VQE and its improved methods[33] are mainly applied in the field of quantum chemistry, aiming to determine the ground state energy of a quantum system. However, when the objective function of combinatorial optimization problems can be expressed as a Hamiltonian operator, the VQE can be used to determine the minimum value of the function. For instance, Carlo *et al.* used the VQE to address resource allocation in a cloud/edge architecture and obtained better results than the QAOA.[34] Zheng *et al.* employed the VQE to solve the perfect matching problem in graph theory, thereby obtaining all perfect matchings or maximum weight perfect matchings.[35]

Figure 1 depicts a sketch of multi-agent decision-making tasks based on Bayesian networks. This process consists of three levels, arranged from bottom to top: the functional inference level, the individual decision-making level, and the collaborative decision-making level. At the functional inference level, large-scale networks are decomposed into multiple functionally independent subnetworks that handle basic inference tasks. The individual decision-making level generates a local optimal strategy (or strategy probability distribution) from the output of the functional inference level. Finally, the collaborative decision-making level integrates individual decision-making and generates optimal strategies using game theory, auction mechanism or consensus algorithms. However, at the functional inference level, existing quantum computers are difficult to directly represent a large-scale network due to the limited number of qubits. Additionally, at the collaborative decision-making level, there are currently no quantum methods available. To address these challenges, this paper proposes a hybrid quantum–classical multi-agent decision-making framework based on hierarchical Bayesian networks in the NISQ era. It comprises two novel methods: a hybrid quantum–classical inference method based on hierarchical Bayesian networks and a VQE-based multi-agent decision-making method in the influence diagram. The first method de-

composes large-scale networks into hierarchical subnetworks. The inference for each subnetwork can be executed on NISQ devices and intermediate results are transformed into classical data for cross-layer transmission. This hierarchical method reduces the depth of the quantum circuit, suppresses the noise accumulation, and improves inference accuracy. The second method models the collaborative decision-making in an influence diagram and aggregates single-agent inferences, encodes the expected utility of diverse actions into a Hamiltonian and efficiently determines the intra-group optimal action. Through this research, we fully utilize the quantum computing resources in the NISQ era to construct a scalable hybrid quantum–classical decision-making framework for multi-agent systems. Our research extends quantum computing to multi-agent decision-making and offers a practical solution in the NISQ era.

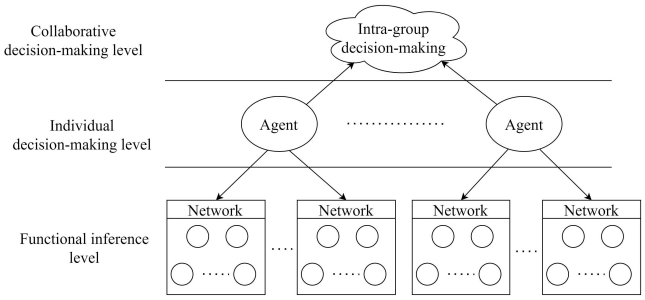


Fig. 1. Sketch of multi-agent decision-making tasks based on Bayesian networks.

This paper is organized as follows. Section 2 briefly introduces Bayesian networks, hierarchical Bayesian networks, influence diagrams, quantum Bayesian networks and VQE, laying the foundation for our proposed method. In Section 3, we design a framework and elaborate on the two methods in detail. In Section 4, we experimentally validate a specific task on the IonQ quantum simulator using the proposed algorithm. Finally, in Section 5, we summarize the research work and outline the future research directions.

2. Background knowledge

This section provides a concise overview of the relevant background knowledge. First, we introduce the fundamental concepts of Bayesian networks, which are the foundation for probabilistic inference under uncertainty. We then introduce hierarchical Bayesian networks, which are an extension of Bayesian networks. Next, we present influence diagrams that incorporate Bayesian networks with decision and utility nodes to determine optimal strategy based on maximum expected theory. Then, we introduce quantum Bayesian networks, which are quantum versions of Bayesian networks. Finally, we introduce VQE. These introductions cover all essential knowledge underlying our proposed method.

2.1. Bayesian networks

The Bayesian network is a probabilistic graphical model.[5] Based on conditional independence assumptions, it effectively represents the joint probability distribution of multiple discrete random variables using a directed acyclic graph (DAG). Figure 2 shows a 7-node Bayesian network, and its main elements include (1) nodes, $X = \{X_1, X_2, \ldots, X_7\}$ represents the set of nodes; (2) edges, directed edges are used to describe the conditional dependence relationships between nodes, which are quantified by the conditional probability table (CPT); and (3) joint probability distribution, its specific form is

$$P(X) = \prod_{i=1}^{7} P(X_i|\text{Parents}(X_i)), \tag{1}$$

where $\text{Parents}(X_i)$ represents the set of parent nodes of X_i.

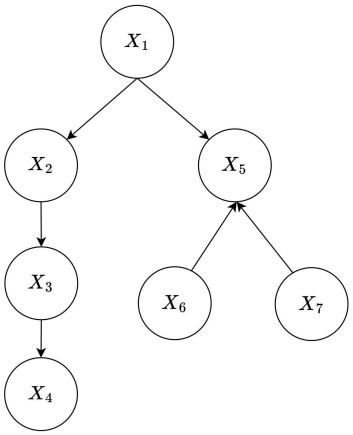


Fig. 2. A 7-node Bayesian network.

There are three types of connection structures in Fig. 2: (1) $X_2 \to X_3 \to X_4$ is the serial connection. Given X_3, then $X_2 \perp X_4$. (2) $X_1 \to \{X_2, X_5\}$ is the diverging connection. Given X_1, then $X_2 \perp X_5$. (3) $\{X_6, X_7\} \to X_5$ is the converging connection. If X_5 is unknown, then $X_6 \perp X_7$.

The inference tasks are mainly to calculate the marginal probability $P(X_i)$ and the conditional probability $P(X_i|e)$ of variables, where e represents the evidence information. This requires determining the conditional independence between nodes according to the D-separation principle.[5]

A Bayesian network is called a polytree if there is exactly one path between any two nodes. Otherwise, it is called a multiply-connected network. The message passing algorithm proposed by Pearl is an efficient exact inference algorithm for the polytree.[36] For multiply-connected networks, the message passing algorithm is not applicable, and the junction tree algorithm is usually used instead.

2.2. Hierarchical Bayesian network

Hierarchical Bayesian networks extend standard Bayesian networks' representational capacity.[37] The network is divided into multiple levels according to the functions of

nodes. Subnetworks at the same level can be processed in parallel, thereby accelerating inference. The Bayesian network in Fig. 2 can be divided into a two-level structure: (1) level 1: $X_2 \to X_3 \to X_4$ and $\{X_6, X_7\} \to X_5$, (2) level 2: $X_1 \to \{X_2, X_5\}$, as shown in Fig. 3.

Here, X_2 and X_5 serve as interface nodes for cross-layer message transmitting. Although subnetworks at the same level are not necessarily independent of each other, they can perform inference independently. Messages are propagated bidirectionally through interface nodes, which connect the subnetworks.

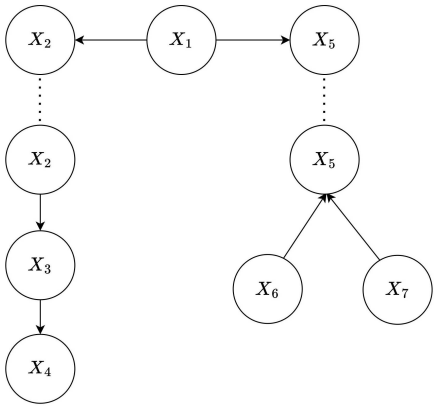


Fig. 3. A 2-level hierarchical Bayesian network.

2.3. Influence diagrams

The influence diagram (decision network) is an extension of the Bayesian network. It introduces decision nodes and utility nodes to support optimal decision-making under uncertain environments.[38] Figure 4 shows a 3-node influence diagram. The nature node represents random variables, and its states are described by probability distributions. The decision node represents the set of optional actions. The utility node is used to quantify the payoffs of different decision-making paths. The utility function takes into account all possible outcomes and their values. In this way, the influence diagram provides decision-makers with a structured method to evaluate the potential impacts of different decisions and select optimal strategies.

The decision-making criterion is based on the maximum expected utility (MEU) principle. For Fig. 4, the formula for calculating the expected utility is

$$EU(A) = \sum_{x_i \in X} P(x_i) \times U(x_i, A), \tag{2}$$

where A represents the set of actions in the decision node, and x_i represents the i-th state of the nature node. The optimal decision is

$$a_{\text{opt}} = \arg\max_a EU(A). \tag{3}$$

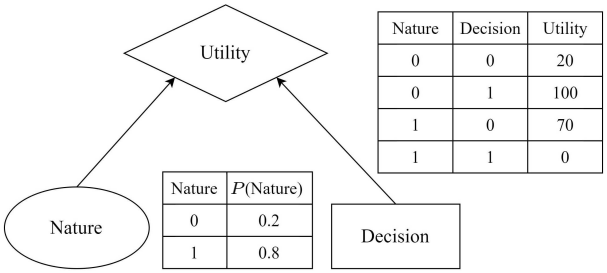


Fig. 4. A 3-node influence diagram.

2.4. Quantum Bayesian networks

The quantum Bayesian network is essentially represented by a quantum superposition state. Figure 5 shows a quantum Bayesian network, which corresponds to $X_1 \to \{X_2, X_5\}$ in Fig. 2. The quantum registers $|X_1\rangle, |X_2\rangle, |X_5\rangle$ correspond to X_1, X_2, X_5, respectively. Assume all nodes are binary (states 0 or 1), each register requires one qubit.

The $Ry(\theta)$ gate can encode probabilities into the amplitudes of quantum states. For instance, when $Ry(\theta_{X_1})$ acts on $|0\rangle$, the quantum state becomes

$$Ry(\theta_{X_1})|0\rangle = \cos\frac{\theta_{X_1}}{2}|0\rangle + \sin\frac{\theta_{X_1}}{2}|1\rangle, \tag{4}$$

where θ_{X_1} is determined by the following formula:

$$\theta_{X_1} = 2 \times \tan^{-1}\sqrt{\frac{P(X_1 = 1)}{P(X_1 = 0)}}. \tag{5}$$

The conditional dependence between X_1 and X_2 can be implemented using two $CRy(\theta)$ gates. When $|X_1\rangle$ is $|0\rangle$, $P(X_2|X_1 = 0)$ can be encoded by applying $CRy(\theta_{X_2,0})$ on $|X_1X_2\rangle$, where $\theta_{X_2,0}$ is determined by the following formula:

$$\theta_{X_2,0} = 2 \times \tan^{-1}\sqrt{\frac{P(X_2 = 1 \mid X_1 = 0)}{P(X_2 = 0 \mid X_1 = 0)}}. \tag{6}$$

The remaining rotation angles can be calculated using the same method. These transformations enable the successful construction of a quantum Bayesian network.

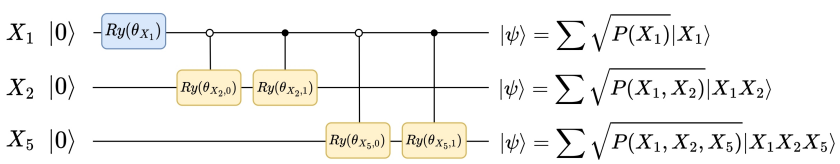


Fig. 5. A 3-node quantum Bayesian network corresponding to $X_1 \to \{X_2, X_5\}$.

2.5. The VQE method

The VQE was initially designed to find the ground state energy of a system's Hamiltonian.[32] For any given objective function that can be represented as a quantum circuit, VQE can be used to find the minimum value of the function. The implementation process is as follows.

(1) Construct an ansatz $|\psi(\theta)\rangle$, where $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$ represents a set of adjustable parameters.

(2) Objective function encoding. The objective function is mapped to a Hamiltonian, typically expressed as a weighted linear combination of Pauli operators

$$H = \sum_i c_i P_i, \quad P_i \in \{I, X, Y, Z\}^{\otimes n}, \tag{7}$$

where c_i is a real number, and P_i is the operator of the i-th qubit.

(3) Parameter optimization. Use a classical optimizer (such as gradient descent) to iteratively update the parameters θ and minimize the expected value of the Hamiltonian. We can obtain the optimization parameter by the following formula:

$$\theta_{\text{opt}} = \arg\min_{\theta} \langle \psi(\theta)|H|\psi(\theta)\rangle . \tag{8}$$

The quantum state $|\psi(\theta_{\text{opt}})\rangle$ encodes the minimum value of the objective function. VQE's shallow circuit design and hybrid quantum–classical architecture make it particularly suitable for NISQ devices.

3. Framework design

In this section, we design the hybrid quantum–classical multi-agent decision-making framework based on hierarchical Bayesian networks, as illustrated in Fig. 6.

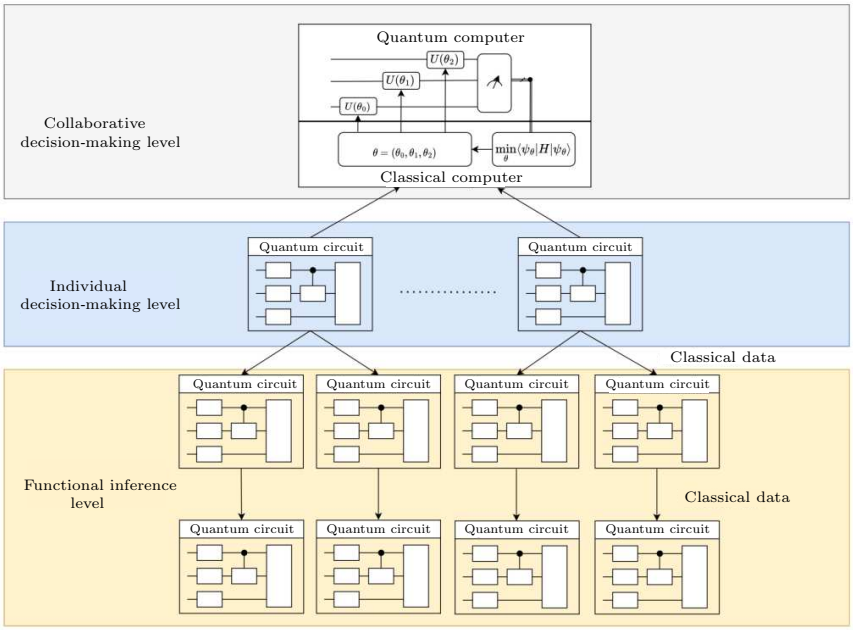


Fig. 6. Framework diagram of hybrid quantum–classical multi-agent decision-making based on hierarchical Bayesian networks.

Our proposed framework comprises three hierarchical levels. At the functional inference level, the network is restructured into a hierarchical Bayesian network and decomposed into multiple NISQ-compatible subnetworks. The inference for each subnetwork can be executed by a quantum circuit. Since different subnetworks exhibit dependent relationships, classical data, which represents these dependencies, is transmitted between distinct quantum circuits. Through the above method, NISQ devices can perform inference tasks at the functional inference level. At the individual decision-making level, functional inference outcomes are encoded into quantum circuits to complete the inference process. At the collaborative decision-making level, we represent the multi-agent decision-making using an influence diagram. In this diagram, an expected utility function is utilized to quantify the value of each action. Using Eqs. (2) and (3), the VQE method determines the optimal strategy.

Next, we elaborate on the two methods corresponding to the functional inference level and the collaborative decision-making level: a hybrid quantum–classical inference method based on hierarchical Bayesian networks and a VQE-based multi-agent decision-making method in the influence diagram.

3.1. Hybrid quantum–classical inference method based on hierarchical Bayesian networks

Figure 7 shows a sketch of the hybrid quantum–classical inference process based on the hierarchical Bayesian network. The network on the left can be decomposed into two-layer

subnetworks. Subnetwork 1 contains nodes $\{X_1, X_2, \ldots, X_L, A\}$ and subnetwork 2 contains nodes $\{A, Y_1, Y_2, \ldots, Y_L, E\}$, where A is an interface node and E is an evidence node. Assume all nodes have n discrete states: $X_l \in \{x_1^{(l)}, \ldots, x_n^{(l)}\}$, $Y_l \in \{y_1^{(l)}, \ldots, y_n^{(l)}\}$, $E \in \{e_1, \ldots, e_n\}$, $A \in \{a_1, \ldots, a_n\}$ for all $l \in \{1, \ldots, L\}$, and the initial probability distribution of the root nodes $\{X_l\}_{l=1}^{L}$ and the CPT are given. The aim is to calculate the prior probabilities $P_q(X_l), P_q(Y_l)$ and posterior probabilities $P_q(X_l|e_k), P_q(Y_l|e_k)$ when $E = e_k$. In the process of the inference task, subnetwork 1 and subnetwork 2 respectively perform inference computation on the quantum processors. The interface A, denoted by dashed lines, is responsible for transmitting classical data between the two quantum circuits.

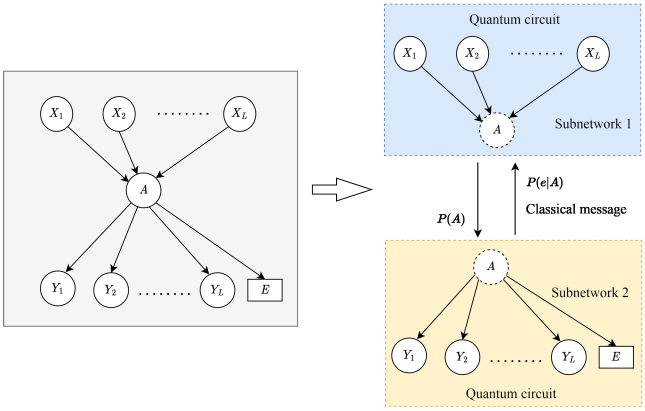


Fig. 7. Schematic diagram of hybrid quantum–classical inference process based on hierarchical Bayesian network.

Next, we detail the method design.

Input Hierarchical Bayesian network: subnetwork 1 $\rightarrow$ subnetwork 2. Subnetwork 1: $\{X_1, X_2, \ldots, X_L\} \rightarrow A$, subnetwork 2: $A \rightarrow \{Y_1, Y_2, \ldots, Y_L, E\}$. A is an interface node, and $E = e_k$ is evidence information. Each node has n states.

Output Prior probabilities $P_q(X_l)$ and $P_q(Y_l)$, posterior probabilities $P_q(X_l|e_k)$ and $P_q(Y_l|e_k)$.

Step 1. Calculate prior probabilities of nodes in subnetwork 1.

As shown in Fig. 8, the quantum registers are $|X_1\rangle|X_2\rangle\cdots|X_L\rangle|A\rangle$, where each register requires $s_n = \lceil \log_2 n \rceil$ qubits. We use the parameterized unitary operators $\{U(\gamma_{X_l})\}_{l=1}^{L}$ to encode the initial probability distribution $\{\pi(X_l)\}_{l=1}^{L}$ into the quantum state. Subsequently, the conditional probabilities $\{P(A|X_l)\}_{l=1}^{L}$ are encoded through the operators $\{V_l(\gamma_{P(A|X_l)})\}_{l=1}^{L}$ to realize the quantum state containing the joint probability distribution $P(X_1, X_2, \ldots, X_L, A)$. Note that in operator V_1, two distinct short lines and their corresponding numbers of qubits are marked, indicating that V_1 only acts on $|X_1\rangle$ and $|A\rangle$, without acting on other registers. The final state denoted by $|\psi_1\rangle$ is

$$|\psi_1\rangle = \sum_{i_1,\ldots,i_L,i_a} \sqrt{P(X_1 = x_{i_1}^{(1)}, \ldots, X_L = x_{i_L}^{(L)}, A = a_{i_a})} \times \left|x_{i_1}^{(1)}\right\rangle \ldots \left|x_{i_L}^{(L)}\right\rangle |a_{i_a}\rangle . \tag{9}$$

Projective measurements on $|A\rangle$ and $|X_l\rangle$ yield the prior distributions

$$P_q(A = a_{i_a}) = \sum_{i_1,\ldots,i_L} |\langle a_{i_a}|\psi_1\rangle|^2 , \tag{10}$$

$$P_q\left(X_l = x_{i_l}^{(l)}\right) = \sum_{i_{-l},i_a} |\langle x_{i_l}^{(l)}|\psi_1\rangle|^2 . \tag{11}$$

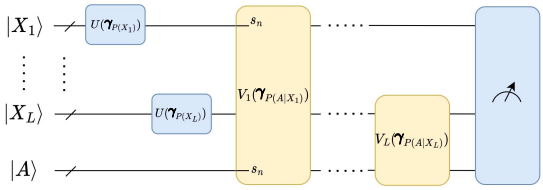


Fig. 8. The quantum circuit calculating the prior probabilities of nodes in subnetwork 1.

Step 2 Calculate the prior probabilities of nodes in subnetwork 2.

As demonstrated in Fig. 9, the quantum registers are $|A\rangle|Y_1\rangle|Y_2\rangle\cdots|Y_L\rangle|E\rangle$, where each register needs $s_n = \lceil \log_2 n \rceil$ qubits. The prior probability $P_q(A)$ obtained from subnetwork 1 is encoded into quantum circuit through the operator $U(\gamma_{P(A)})$. We encode the conditional probabilities $\{P(Y_l|A)\}_{l=1}^{L}$ and $P(E|A)$ into the quantum state through the operators $\{V_l(\gamma_{P(Y_l|A)})\}_{l=1}^{L}$ and $V_{L+1}(\gamma_{P(E|A)})$. The specific form of the final state $|\psi_2\rangle$ is

$$|\psi_2\rangle = \sum_{i_a,i_1,\ldots,i_L,i_e} \sqrt{P_q(A = a_{i_a})} \prod_{l=1}^{L} \sqrt{P(Y_l = y_{i_l}^{(l)}|A = a_{i_a})} \times \sqrt{P(E = e_{i_e}|A = a_{i_a})}|a_{i_a}\rangle|y_{i_1}^{(1)}\rangle\cdots|y_{i_L}^{(L)}\rangle|e_{i_e}\rangle. \tag{12}$$

Perform a projective measurement on $|Y_l\rangle$ respectively to obtain the prior distribution

$$P_q\left(Y_l = y_{i_l}^{(l)}\right) = \sum_{i_{-l},i_a} |\langle y_{i_l}^{(l)}|\psi_2\rangle|^2 . \tag{13}$$

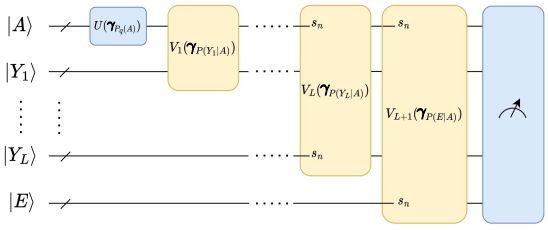


Fig. 9. The quantum circuit calculating the prior probabilities of nodes in subnetwork 2.

Step 3 Calculate the likelihood function $P_q(e_k|A)$ in subnetwork 2.

As exhibited in Fig. 10, the quantum registers are still $|A\rangle|Y_1\rangle|Y_2\rangle\cdots|Y_L\rangle|E\rangle$. Hadamard gates are applied to $|A\rangle$ instead of $U(\gamma_{P(A)})$. And the rest of the quantum circuit is the same as Fig. 9. So, the final state $|\psi_3\rangle$ is

$$|\psi_3\rangle = \sum_{i_a,i_1,\ldots,i_L,i_e} \frac{1}{\sqrt{n}} \prod_{l=1}^{L} \sqrt{P(Y_l = y_{i_l}^{(l)}|A = a_{i_a})}$$

$$\times\sqrt{P(E=e_{i_e}|A=a_{i_a})}|a_{i_a}\rangle|y_{i_1}^{(1)}\rangle\cdots|y_{i_L}^{(L)}\rangle|e_{i_e}\rangle. \quad (14)$$

Post-selection on measurements of all registers with $E=e_k$ yields the likelihood function

$$P_q(e_k|A=a_{i_a})=n\times\sum_{i_1,\ldots,i_L}\left|\left\langle a_{i_a},y_{i_1}^{(1)}\cdots y_{i_L}^{(L)},e_k\,\middle|\,\psi_3\right\rangle\right|^2. \quad (15)$$

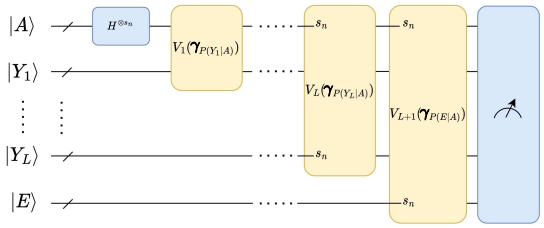


Fig. 10. The quantum circuit calculating the likelihood function in subnetwork 2.

Step 4. Calculate posterior probability $P_q(X_l|e_k)$ in subnetwork 1.

As shown in Fig. 11, based on the quantum circuit in Fig. 8, a quantum register $|A_L\rangle$ is added, which requires $s_n=\lceil\log_2 n\rceil$ qubits. We encode the likelihood function $P_q(e_k|A)$ obtained from subnetwork 2 into the quantum state through the operator $U(\gamma_{P_q(e_k|A)})$ acting on $|A_L\rangle$, and the other operators are consistent with those in Fig. 8. The quantum system evolves to the final state $|\psi_4\rangle$,

$$\begin{aligned}|\psi_4\rangle=&\sum_{i_1,\ldots,i_L,i_a}\sqrt{P(X_1=x_{i_1}^{(1)},\ldots,X_L=x_{i_L}^{(L)},A=a_{i_a})}\\&\times|x_{i_1}^{(1)}\rangle\cdots|x_{i_L}^{(L)}\rangle|a_{i_a}\rangle\otimes\sum_{j_a}\sqrt{P(e_k|A=a_{j_a})}\,|a_{j_a}\rangle. \quad (16)\end{aligned}$$

We measure all the registers and calculate the posterior probability

$$P_q(X_l=x_{i_l}^{(l)}|e_k)=\sum_{x_{-l},i_a}\left|\left\langle x_{i_1}^{(1)},\ldots,x_{i_L}^{(L)},a_{i_a},a_{i_a}\,\middle|\,\psi_4\right\rangle\right|^2. \quad (17)$$

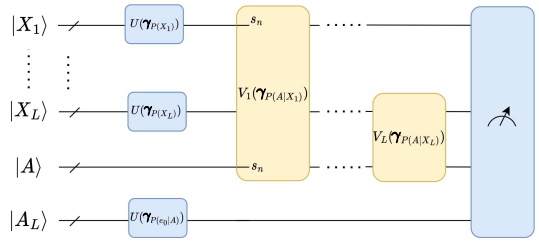


Fig. 11. The quantum circuit calculating the posterior probabilities of nodes in subnetwork 1.

Step 5 Update posterior probability $P_q(Y_l|e_k)$ in subnetwork 2.

As depicted in Fig. 12, we extend the circuit in Fig. 9 by adding register $|A_L\rangle$. This register requires $s_n=\lceil\log_2 n\rceil$ qubits. We encode the likelihood function $P_q(e_k|A)$ on $|A_L\rangle$, and the construction of the remaining component is consistent with that in Fig. 10. The final state denoted by $|\psi_5\rangle$ is

$$\begin{aligned}|\psi_5\rangle=&\sum_{i_a,i_1,\ldots,i_L}\frac{1}{\sqrt{n}}\prod_{i=1}^{L}\sqrt{P(Y_l=y_{i_l}^{(l)}|A=a_{i_a})}\\&\times\sqrt{P(E=e_k|A=a_{i_a})}|a_{i_a}\rangle|y_{i_1}^{(1)}\rangle\cdots|y_{i_L}^{(L)}\rangle|e_k\rangle\\&\otimes\sum_{j_a}\sqrt{P(e_k\,|\,A=a_{j_a})}|a_{j_a}\rangle. \quad (18)\end{aligned}$$

We measure all registers and obtain the posterior probability

$$\begin{aligned}&P_q(Y_l=y_{i_l}^{(l)}|e_k)\\&=n\times\sum_{i_a,i_{-l}}\left|\left\langle a_{i_a},y_{i_1}^{(1)},\ldots,y_{i_L}^{(L)},e_k,a_{i_a}\,\middle|\,\psi_5\right\rangle\right|^2. \quad (19)\end{aligned}$$

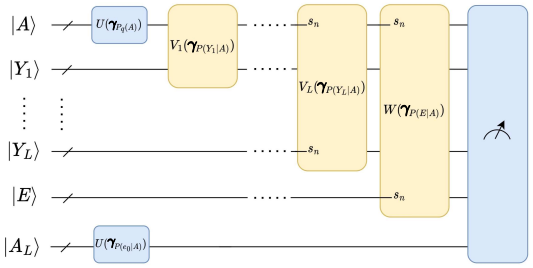


Fig. 12. The quantum circuit updating the posterior probabilities of nodes in subnetwork 2.

Through the above five steps, we have successfully completed the two-layer Bayesian network inference task. This method uses quantum computing to calculate prior probabilities, posterior probabilities, and likelihood functions. Classical data including prior probabilities and likelihood functions are exchanged between circuits.

3.2. VQE-based multi-agent decision-making method in the influence diagram

Figure 13 shows an influence diagram composed of one leader agent and n follower agents. The leader agent dominates the decision-making process. All agents have the same actions, and the action set is $A=\{a_1,a_2,\ldots,a_m\}$. Let the i-th action of the leader be $a_i^{\mathrm{L}}\in A$, its probability distribution is $P(a_i^{\mathrm{L}})$. The i-th action of the j-th follower is $a_i^{\mathrm{F}_j}\in A$ and its probability distribution is $P(a_i^{\mathrm{F}_j})$. The i-th action of the intra-group decision-making node is $a_i^{\mathrm{G}}\in A$, where $a_i^{\mathrm{G}}\in\{0,1\}$. When $a_i^{\mathrm{G}}=1$, the action is executed. Otherwise, the action is not executed. The actions are mutually exclusive:

$$\sum_{i=1}^{m}a_i^{\mathrm{G}}=1. \quad (20)$$

To balance the leader's dominance and the group synergy effect, we define a joint utility function whose form is

$$\begin{aligned}&U_G(a_i^{\mathrm{G}},a_j^{\mathrm{L}},a_{j_1}^{\mathrm{F}_1},\ldots,a_{j_n}^{\mathrm{F}_n})\\&=w_L U(a_j^{\mathrm{L}})+\sum_{i=1}^{N}w_{\mathrm{F}_i}U(a_{j_i}^{\mathrm{F}_i})+\alpha S(a_i^{\mathrm{G}},a_j^{\mathrm{L}},a_{j_1}^{\mathrm{F}_1},\ldots,a_{j_n}^{\mathrm{F}_n}), \quad (21)\end{aligned}$$

where $U(\cdot)$ represents the utility value when a single agent selects an action. w_{L} and w_{F_i} are the decision-making weights of the leader and the followers, respectively. With $w_{\mathrm{L}} > w_{\mathrm{F}_i}$, α represents the synergy utility function measuring action complementarity among agents.

For an intra-group action a_i^{G} , the formula for calculating the expected utility is

$$EU(a_i^{\mathrm{G}}) = \sum_{a_j^{\mathrm{L}}, a_{j_1}^{\mathrm{F}_1}, \ldots, a_{j_n}^{\mathrm{F}_n}} P(a_j^{\mathrm{L}}) \prod_{l=1}^{n} P(a_{j_l}^{\mathrm{F}_l}) U(a_i^{\mathrm{G}}, a_j^{\mathrm{L}}, a_{j_1}^{\mathrm{F}_1}, \ldots, a_{j_n}^{\mathrm{F}_n}). \quad (22)$$

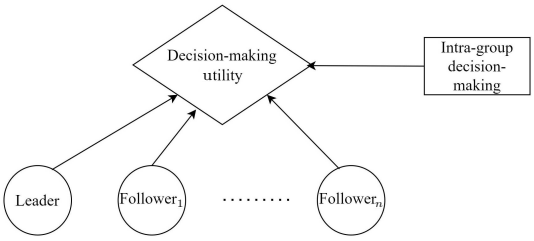


Fig. 13. The influence diagram composed of a leader–follower multi-agent system.

The optimal action with maximum expected utility is determined using VQE. The key steps of the VQE in solving combinatorial optimization problems are to convert the mathematical model into a quadratic unconstrained binary optimization (QUBO) problem and then further map it onto the Ising model.[39] Therefore, we first convert the maximum expected utility into a minimized QUBO problem

$$\min f = -\sum_{i=1}^{m} EU(a_i^{\mathrm{G}}) a_i^{\mathrm{G}} + \lambda \left(\sum_{i=1}^{m} a_i^{\mathrm{G}} - 1 \right)^2, \quad (23)$$

where λ is the penalty coefficient, $\lambda > 0$. It depends on the form of the objective function. Commonly, it can be chosen as

$$\lambda = \sum_{i=1}^{m} EU(a_i^{\mathrm{G}}) + 1. \quad (24)$$

This setting for the penalty coefficient ensures that the penalty equals zero only if a_i^{G} satisfies all the constraints. If any constraint is violated, the penalty value must be greater than or equal to λ. Since the penalty value is always greater than the possible maximum value of the first term in Eq. (23), it guarantees that the cost of violating constraints is higher than the intrinsic value of the objective function itself, thereby ensuring that the minimum solution of the target f is found when the constraints are satisfied.

Next, we convert the QUBO problem to the Ising model. In the QUBO problem, the decision variables are binary, taking the values 0 or 1. However, in the Ising model, the decision variables are usually represented as $+1$ or -1. To achieve this conversion, let

$$a_i^{\mathrm{G}} = \frac{1 - \sigma_Z^i}{2}. \quad (25)$$

Therefore, the Hamiltonian can be obtained as

$$H = -\sum_{i=1}^{m} EU(a_i^{\mathrm{G}}) \frac{1 - \sigma_Z^i}{2} + \lambda \left(\sum_{i=1}^{m} \frac{1 - \sigma_Z^i}{2} - 1 \right)^2. \quad (26)$$

Then let's take $m = 4$ as an example to construct the ansatz $|\psi(\theta)\rangle$. We can utilize the TwoLocal circuit in Qiskit, which is a parametric circuit composed of alternating rotation layers and entanglement layers. The rotation layer can choose the single-qubit rotation gate $Ry(\theta)$, and the entanglement layer can select the simplest CNOT gate, repeated r times. The ansatz circuit is shown in Fig. 14.

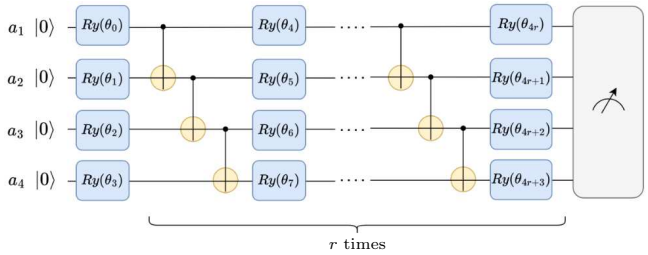


Fig. 14. The ansatz $|\psi(\theta)\rangle$ when $m = 4$ in the VQE method.

A classical optimizer minimizes the expectation value $\langle\psi(\theta)|H|\psi(\theta)\rangle$ to obtain the optimal parameter θ_{opt}. Encode θ_{opt} into the parameterized quantum circuit and perform the measurement. We can obtain the optimal action $a_{\mathrm{opt}}^{\mathrm{G}}$ by selecting the quantum state with the largest probability amplitude.

4. Experiments

In this section, we first describe a specific task in a multi-agent defense system in a marine battlefield environment. Subsequently, we model the multi-agent decision-making tasks as a hierarchical Bayesian network. Thereafter, we validate the feasibility and effectiveness of the proposed method. This section is partitioned into four parts: task description, modeling of multi-agent decision-making tasks, experimental parameter settings and result analysis.

4.1. Task description

In the marine battlefield environment, our side deploys a multi-agent system for monitoring and defense. The system consists of a leader agent and multiple follower agents. The leader agent carries high-precision sensors, whereas followers carry lower-precision ones. When an enemy threat source enters our marine area, the multi-agent system can efficiently respond to an enemy threat and achieve autonomous decision-making. The decision-making process comprises four stages:

(1) The first stage. Each single agent identifies the type and status of the threat source and assesses its own operational capability.

(2) The second stage. Each single agent comprehensively assesses the threat level and operational superiority based on the results of the first stage.

(3) The third stage. According to the results of the second stage, each single agent obtains the probability distribution of the next action.

(4) The fourth stage. The group integrates the decisions of each agent and determines the optimal strategy within the group.

4.2. Modeling of multi-agent decision-making tasks

According to the four-stage decision-making process, we model the task as a hierarchical Bayesian network, as shown in Fig. 15. Different colors distinguish the network layers. Dotted-line nodes represent interfaces, connected by solid lines indicating classical message transmission.

We have simplified the task as follows:

(1) The multi-agent system has three agents, including one leader agent and two follower agents.

(2) The three agents share a unified network topology, and the sensor's accuracy is reflected in the conditional probability between nodes.

(3) The network structure is simple, preserving only a few key nodes and states.

The network is divided into 1–4 layers corresponding to 1–4 decision stages, whose specific topology and node parameters are shown in Table 1. Node conditional dependencies are implemented in Python code.

In the fourth stage, some parameters associated with the synergy utility function need to be configured.

(1) The decision-making weight. The leader agent is 0.75, and the two follower agents are both 0.15.

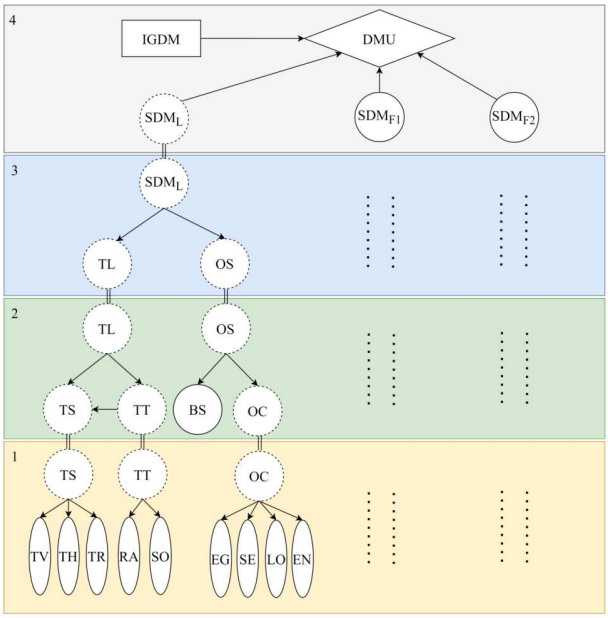


Fig. 15. A 3-agent hierarchical decision-making network.

Table 1. Hierarchical Bayesian network topology and node parameter table.

Layer	Subnetwork	Node name	Node type	Interface	States
1	Threat source status	Threat status (TS)	Nature	✓	{Attack, Recon}
		Threat velocity (TV)	Evidence		{High, Medium, Low}
		Threat heading (TH)	Evidence		{Direct, Detour, Retreat}
		Threat range (TR)	Evidence		{Close, Medium, Far}
	Threat source type	Threat type (TT)	Nature	✓	{USV, UUV}
		Radar (RA)	Evidence		{High, Medium, Low}
		Sonar (SO)	Evidence		{High, Medium, Low}
	Operational capability assessment	Operational capability (OC)	Nature	✓	{Strong, Weak}
		Engine (EG)	Evidence		{Good, Fair, Average}
		Sensor (SE)	Evidence		{Good, Average, Poor}
		Load (LO)	Evidence		{Good, Average, Poor}
		Energy (EN)	Evidence		{Good, Average, Poor}
2	Threat level assessment	Threat level (TL)	Nature	✓	{High, Low}
		Threat status (TS)	Nature	✓	{Attack, Recon}
		Threat type (TT)	Nature	✓	{USV, UUV}
	Operational superiority assessment	Operational superiority (OS)	Nature	✓	{Superiority, Inferiority}
		Battlefield situation (BS)	Evidence		{Clear, Stormy, Night, Jamming}
		Operational capability (OC)	Nature	✓	{Strong, Weak}
3	Individual decision-making	Single-agent decision-making (SDM)	Nature		{Attack, Monitor}
		Threat level (TL)	Nature	✓	{High, Low}
		Operational superiority (OS)	Nature	✓	{Superiority, Inferiority}
4	Collaborative decision-making	Single-agent decision-making (SDM)	Nature		{Attack, Monitor}
		Intra-group decision-making (IGDM)	Decision		{Attack, Monitor}
		Decision-making utility (DMU)	Utility		

(2) The decision-making utility value for a single agent is

$$U(a)=\begin{cases}30, & \text{if} \quad a=\text{attack},\\ 10, & \text{if} \quad a=\text{monitor}.\end{cases} \tag{27}$$

(3) The synergy coefficient is $\alpha=0.5$.

(4) The specific form of the synergy utility function

$$S(\cdot)=\begin{cases}3U(a_i^{\mathrm{G}}), & \text{if } a_i^{\mathrm{G}}=a_j^{\mathrm{L}}\wedge a_i^{\mathrm{G}}=a_{j_1}^{\mathrm{F}_1}=a_{j_2}^{\mathrm{F}_2},\\ 2U(a_i^{\mathrm{G}}), & \text{if } a_i^{\mathrm{G}}=a_j^{\mathrm{L}}\wedge (a_j^{\mathrm{L}}=a_{j_1}^{\mathrm{F}_1}\vee a_j^{\mathrm{L}}=a_{j_2}^{\mathrm{F}_2}),\\ U(a_i^{\mathrm{G}}), & \text{if } a_i^{\mathrm{G}}=a_j^{\mathrm{L}}\wedge \text{other},\\ U(a_i^{\mathrm{G}}), & \text{if } a_i^{\mathrm{G}}\neq a_j^{\mathrm{L}}\wedge a_i^{\mathrm{G}}=a_{j_1}^{\mathrm{F}_1}=a_{j_2}^{\mathrm{F}_2},\\ 0, & \text{if } a_i^{\mathrm{G}}\neq a_j^{\mathrm{L}}\wedge (a_i^{\mathrm{G}}=a_{j_1}^{\mathrm{F}_1}\vee a_i^{\mathrm{G}}=a_{j_2}^{\mathrm{F}_2}),\\ -3U(a_i^{\mathrm{G}}), & \text{otherwise}.\end{cases} \tag{28}$$

4.3. Experimental parameter settings

We select Qiskit as the programming platform for quantum algorithms. Experiments are conducted on both noiseless and noisy quantum simulators, all implemented on classical computers. The computer configuration is Intel Core i9-12900KF 3.19 GHz 32GB RAM. The noise simulators (Aria-1, Aria-2 and Forte-1) emulate actual noise characteristics of IonQ quantum computers. The error rate parameters used for one- and two-qubit gates are shown in Table 2. Each experiment uses 10240 shots to ensure result reliability.

Table 2. Comparison of τ_{1q} and τ_{2q} values for different systems.

System	τ_{1q}	τ_{2q}
Aria-1	0.0005	0.0133
Aria-2	0.000657	0.01856
Forte-1	00.000267	0.00495

To mitigate noise errors, we use IonQ simulator's built-in debiasing technique — compilation-level error mitigation method that effectively reduces the influence of systematic noise by generating multiple equivalent circuit variants and averaging the measurement outcomes. This method does not require additional qubits or shots, and offers the advantage of low latency. However, in certain scenarios, relying only on debiasing may still result in significant residual errors. To address this limitation, we further incorporate the zero-noise extrapolation (ZNE) technique, which involves running the same quantum circuit at different noise levels and extrapolating the obtained expectation values to estimate the ideal noise-free observable. ZNE is one of the simplest and most widely used error mitigation techniques, with the key advantage of not requiring precise knowledge of the underlying noise model parameters. Nevertheless, this method typically leads to an increase in circuit depth, thereby introducing additional resource overhead.

For classical inference benchmarking, we employ Netica software[40] for Bayesian networks and influence diagrams.

In the process of using VQE to solve the multi-agent decision-making problem, the ansatz is a TwoLocal circuit with 1 repetition. The single qubit rotation gate is the *Ry* gate, and the entanglement gate is the *CZ* gate.

Our experiment consists of two parts.

(1) Individual inference task. To verify the effectiveness of the hybrid quantum–classical inference algorithm based on hierarchical Bayesian networks, we will compare the prior probabilities and posterior probabilities of four methods: the noiseless quantum hierarchical method, the noisy quantum hierarchical method, the noisy quantum non-hierarchical method, and the classical method.

(2) Collaborative decision-making task. To verify the effectiveness of the VQE method, we will compare the results of three types of methods: the ideal VQE method, the noisy VQE method, and the classical method. To avoid the influence of quantum noise interference in the first three stages (individual decision-making) on the evaluation of multi-agent intra-group decision-making, this experiment uses the action probability distribution generated by Netica as the input benchmark for the fourth stage. This ensures an independent evaluation of the ability of the VQE method to solve multi-agent collaborative decision-making tasks.

4.4. Experimental results and analysis

Figure 16 presents a comparative analysis of prior probability errors between quantum methods under various conditions and Netica results. In this experiment, the subnetworks involved include the threat level assessment, the threat source status and the threat source type. The X axis denotes the node name, while the Y axis represents the absolute probability error between quantum methods and Netica when the node state is 0. Under noiseless conditions, both quantum hierarchical and non-hierarchical schemes achieve near-zero error margins ($\varepsilon\approx 0$). In the three noisy simulators (Aria-1, Aria-2, Forte-1), the effects of the two schemes are also very similar. Notably, the Forte-1 simulator maintains very small errors ($\varepsilon<0.02$). Both quantum methods show acceptable deviations ($\varepsilon<0.06$) from Netica results.

Two key factors contribute to the low error. (1) Minimal quantum circuit depth. In this experiment, due to the simple network structure and the small number of node states, the quantum circuit only consists of a small number of single-qubit gates and two-qubit gates, significantly reducing susceptibility to gate noise and quantum decoherence effects. (2) Restricted qubit sampling. When calculating the prior probability, sampling is only performed on the qubits corresponding to a single node, which is less affected by noise, so the results are relatively accurate.

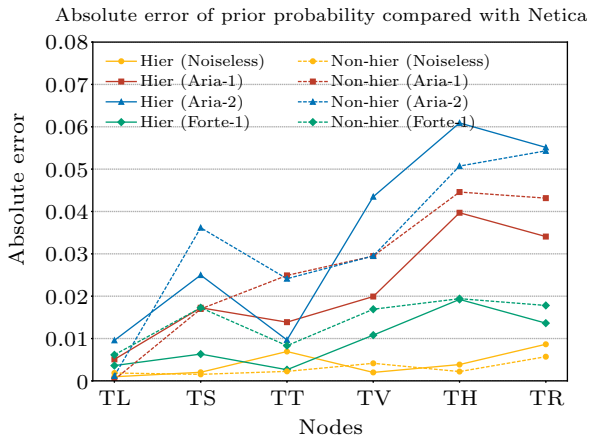


Fig. 16. The comparison of prior probability errors between quantum methods under various conditions and Netica.

Figure 17 presents a comparison of posterior probability errors among two quantum methods under various conditions with Netica results. In this experiment, the subnetworks involved are also the threat level assessment, the threat source status and the threat source type. Each subplot corresponds to a specific noise model configuration with the X axis denoting eight randomly selected evidence cases and the Y axis representing the absolute probability error for the state value "High" of node TL.

Under noiseless conditions, the quantum hierarchical method achieves exceptional accuracy ($\varepsilon \approx 0$), validating the algorithmic correctness. In contrast, the non-hierarchical method exhibits larger average errors.

In the three noisy simulators (Aria-1, Aria-2 and Forte-1), both the quantum methods show posterior probability errors. Notably, Forte-1 demonstrates the best performance with the lowest average error, suggesting its noise model is less disruptive to the quantum computations. In contrast, Aria-2 yields the highest average error ($\varepsilon \approx 0.1$) due to its more challenging noise characteristics. An error of more than 5% can easily lead to incorrect conclusions. The calculation of posterior probabilities involves obtaining samples containing evidence via post-selection methods, which are notably sensitive to noise fluctuations. Consequently, under the same noisy conditions, the error of the posterior probability is significantly larger than that of the prior probability.

Comparatively, the hierarchical method demonstrates smaller errors than its non-hierarchical counterpart. This advantage lies in the fact that the hierarchical method has a lower complexity in quantum circuits and offers better noise resistance. However, certain cases using the hierarchical method under Aria-1 and Aria-2 still exhibit relatively high inference errors.

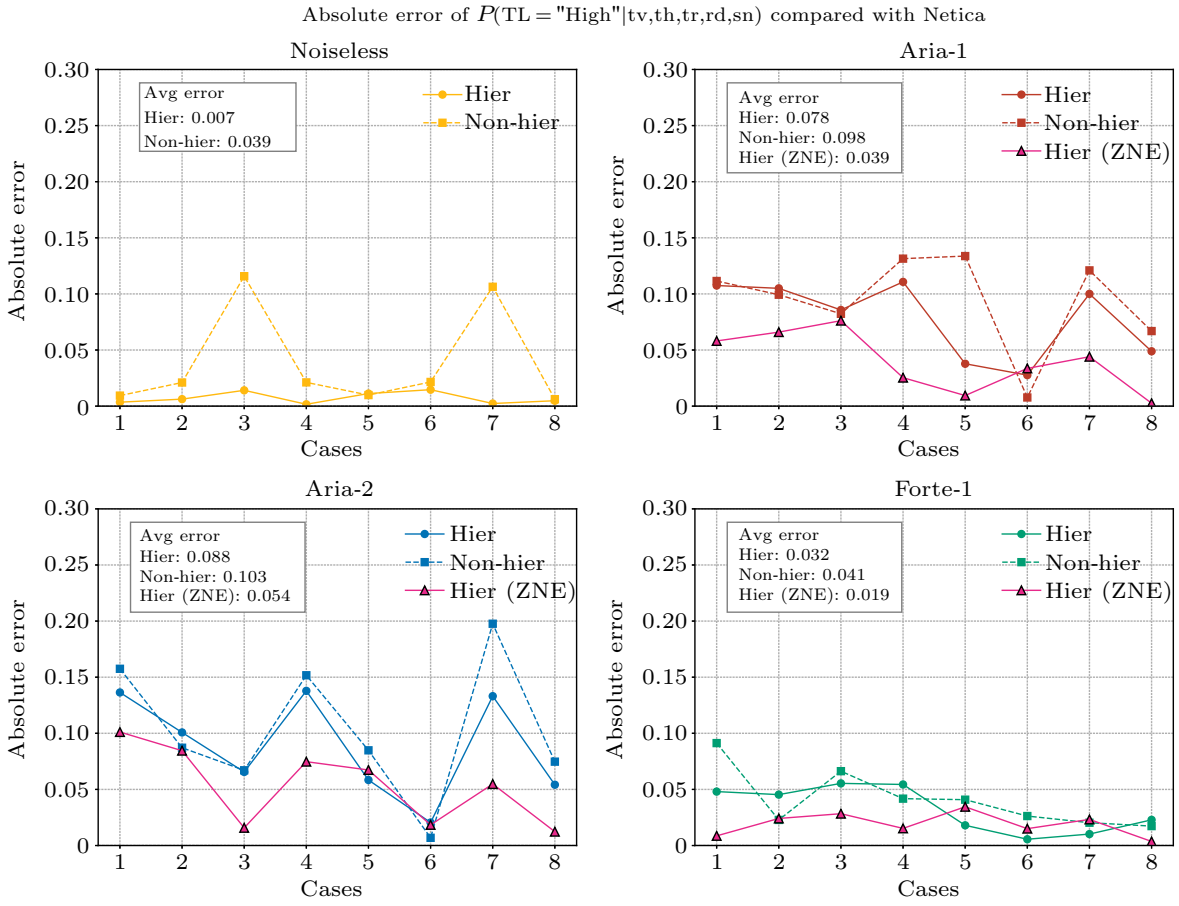


Fig. 17. The comparison of posterior probability errors between quantum methods under various conditions and Netica.

To mitigate these errors, we incorporated ZNE into our framework. The ZNE enhances posterior probability and likelihood function computations, improving information propagation accuracy across subnetwork layers. As shown in Fig. 17, the integration of ZNE effectively reduces the average error by approximately 50%. However, this enhancement comes at the cost of increased circuit depth, which becomes approximately 3 to 5 times deeper compared to the original circuits without ZNE.

Figure 18 presents a comparison of the error in the posterior probabilities $P(SDM|e)$ in the individual decision-making subnetwork between the quantum hierarchical method and Netica, where $e = (\mathrm{tv}, \mathrm{th}, \mathrm{tr}, \mathrm{rd}, \mathrm{sn}, \mathrm{bs}, \mathrm{engine}, \mathrm{sensor}, \mathrm{load}, \mathrm{energy})$. In this experiment, the subnetworks encompass all subnetworks in layers 1–3. The X axis displays eight randomly selected evidence cases. The Y axis denotes the absolute value of the probability error between the quantum method and Netica when the state value of SDM is "Attack". The error of the noiseless quantum hierarchical method is nearly 0. In the three noisy simulators (Aria-1, Aria-2 and Forte-1), Forte-1 demonstrates the best performance with the lowest average error ($\varepsilon \approx 0.03$), and Aria-1 performs second-best ($\varepsilon \approx 0.06$), while Aria-2 shows the worst performance ($\varepsilon \approx 0.08$). In particular, the errors of these simulators on cases 1–4 are extremely large. Therefore, we have added the ZNE module. The results demonstrate significant error reduction for both Aria-1 and Aria-2, while Forte-1 exhibits minimal change. For networks with single-agent decision-making, non-hierarchical methods are unable to perform reasoning, whereas hierarchical methods demonstrate superior results.

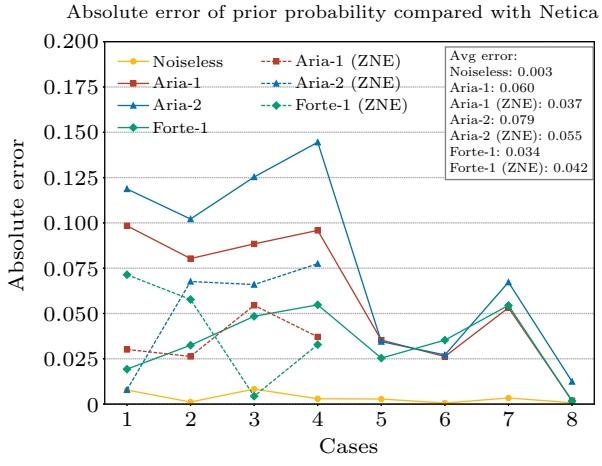


Fig. 18. The error comparison of single-agent decision-making.

Figure 19 shows the performance of multi-agent intra-group decision-making based on VQE and Netica. The X axis represents 8 randomly selected cases, and the Y axis represents the maximum expected utility. A (Attack) and M (Monitor) are marked on the top of the columns, representing the optimal action within the group. Under noiseless conditions, VQE achieves perfect agreement with Netica in maximum expected utility and the action agreement rate reaches 100%, verifying the correctness and rationality of this method. Among the three noisy simulators, Forte-1 exhibits the lowest noise level, and consequently, its maximum expected utility value is very close to that obtained using Netica, followed closely by Aria-1. The largest deviation is observed in Aria-2, where the average error rate reaches up to 15%. Despite this, the action agreement rate remains at 100%. This is because quantum noise affects the computation of expected utilities for all candidate actions in a roughly equivalent manner. Since the selection of the optimal action depends only on the relative ranking of the expected utilities rather than their absolute values, the noise errors do not alter the optimality of the chosen action.

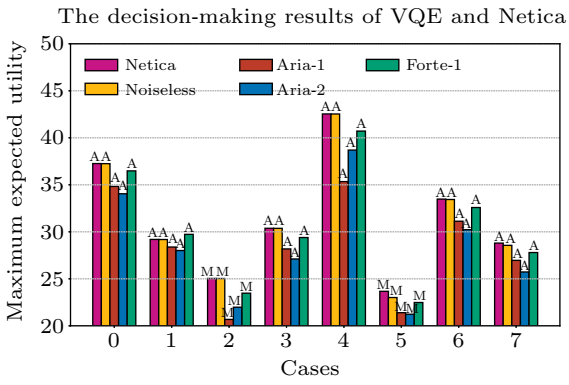


Fig. 19. The decision-making results based on VQE and Netica.

Figure 20 presents the convergence curves of three ansatz circuits (TwoLocal, EfficientSU2, RealAmplitudes) under different noise models. Under noiseless conditions, TwoLocal and RealAmplitudes demonstrate similar convergence rates, achieving stability around the 12th iteration. This indicates their excellent performance and stability. In contrast, EfficientSU2 shows poorer convergence, requiring over 30 iterations to stabilize. This behavior may be attributed to its more complex circuit structure compared to the other two ansatzes, which can lead to local optima during optimization.

In noisy simulators, TwoLocal maintains good convergence despite initial fluctuations, stabilizing after approximately 10 iterations. This demonstrates its inherent noise robustness. EfficientSU2's performance improves significantly in Aria-1 and Forte-1, exhibiting accelerated convergence. There are two potential factors. (1) Noise-induced stochastic perturbations may help escape local minima, facilitating global optimization. (2) The ansatz's complex structure may enhance search capability in noisy environments. RealAmplitudes also demonstrates convergence in noisy conditions, though with greater fluctuation, ultimately reaching low energy values. While showing some noise robustness, its stability proves inferior to TwoLocal, particularly in Aria-1 and Aria-2.

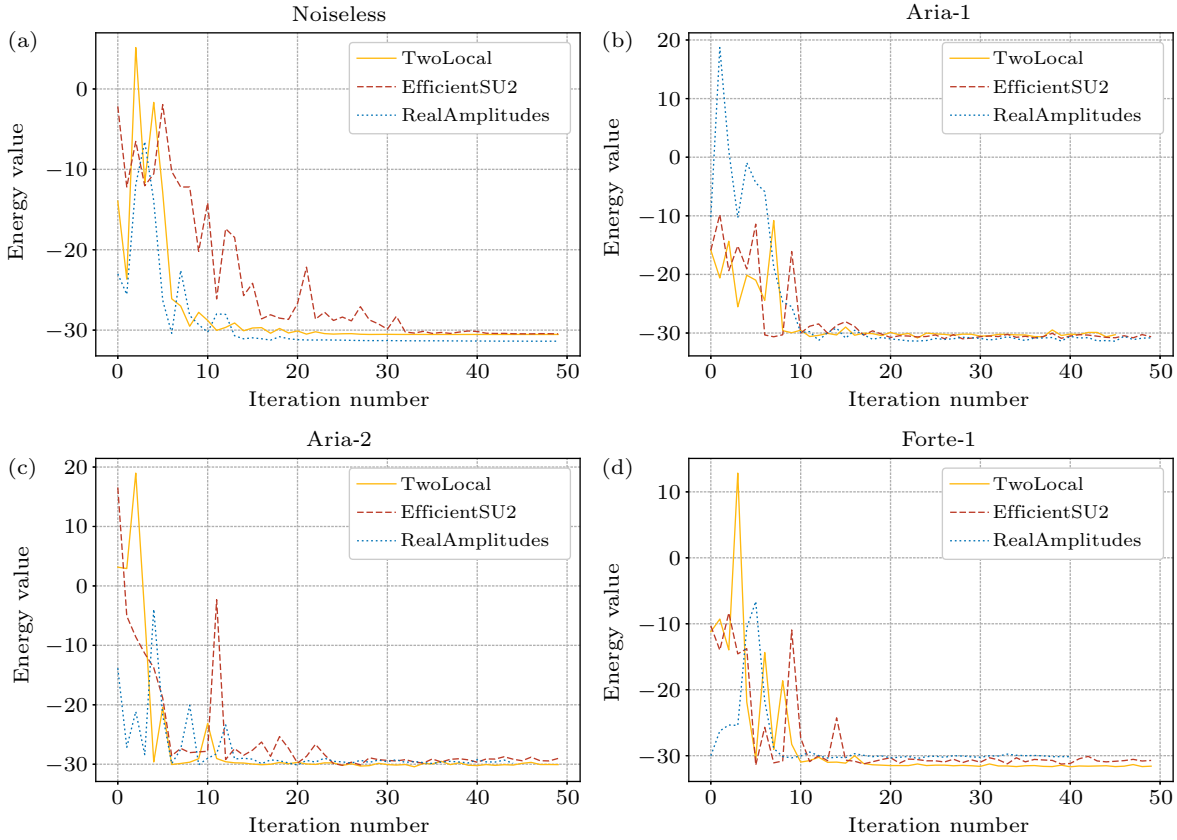


Fig. 20. Comparative analysis of VQE convergence behavior among different ansatz circuits across noise models.

5. Conclusion

To address the tension between the decision-making of large-scale Bayesian networks in multi-agent systems and the restricted quantum resources on NISQ devices, and considering that there are currently no quantum methods available for collaborative decision-making. This paper proposes a hybrid quantum–classical multi-agent decision-making framework based on hierarchical Bayesian networks in the NISQ era. It comprises two novel methods. The first one is a hybrid quantum–classical inference method based on hierarchical Bayesian networks for individual decision-making. This method effectively reduces both quantum circuit depth and noise susceptibility. Compared with the non-hierarchical method, it demonstrates enhanced inference accuracy. The second one is a VQE-based method in the influence diagram for collaborative decision-making. This method can determine the optimal action within the group efficiently. This work not only presents new perspectives for the application of quantum computing in the field of multi-agent autonomous decision-making but also delivers a practical NISQ-era implementation solution.

Future research will focus on further exploring hybrid quantum–classical inference and decision-making tasks in multi-agent systems. For instance, we will explore whether the incorporation of quantum communication can enhance the collaborative inference and decision-making abilities among agents. We will also extend the hierarchical framework to dynamic Bayesian networks for temporal inference tasks on NISQ devices. Furthermore, we will investigate how quantum computing addresses game-theoretic issues that arise in multi-agent decision-making.

Acknowledgment

Project supported by the National Natural Science Foundation of China (Grant Nos. 62473371 and 61673389).

References

[1] Buskey G, Wyeth G and Roberts J 2001 *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 21–26, 2001, Seoul, Korea, p. 1635
[2] Wu J H, Zhang T, Li Y and Zhou G J 2023 *Expert Systems with Applications* **240** 122448
[3] Slowinski R, Greco S and Matarazzo B 2022 *Rough Sets in Decision-Making* (Berlin, Heidelberg: Springer) pp. 1
[4] Ni Z, Phillips L D and Hanna G B 2010 *The Use of Bayesian Networks in Decision-Making* (Berlin, Heidelberg: Springer) pp. 351
[5] Koller D and Friedman N 2009 *Probabilistic Graphical Models: Principles and Techniques* (Cambridge, MA, USA: MIT Press)
[6] Neil M, Fenton N and Nielsen L 2000 *The Knowledge Engineering Review* **15** 257
[7] Lin C S and Garrido M I 2022 *Neuroscience & Biobehavioral Reviews* **137** 104649
[8] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge: Cambridge University Press)
[9] Shor P 1994 *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (SFCs'94)*, November 20–22, 1994, Washington, DC, United States, p. 124

[10] Grover L K 1996 *Proceedings of the 28th ACM Symposium on Theory of Computing (STOC'96)*, May 22–24, 1996, Philadelphia, Pennsylvania, USA, p. 212
[11] Harrow A W, Hassidim A and Lloyd S 2009 *Phys. Rev. Lett.* **103** 150502
[12] Preskill J 2018 *Quantum* **2** 79
[13] Blekos K, Brand D, Ceschini A, *et al.* 2024 *Physics Reports* **1068** 1
[14] Cerezo M *et al.* 2021 *Nature Reviews Physics* **3** 625
[15] Farhi E, Goldstone J and Gutmann S 2014 arXiv:1411.4028 [quant-ph]
[16] Tucci R R 1995 *International Journal of Modern Physics B* **9** 295
[17] Low G H, Yoder T J, Chuang I L 2014 *Phys. Rev. A* **89** 062315
[18] Biamonte J D, Wittek P, Pancotto N, *et al.* 2017 *Nature* **549** 195
[19] Jin Y X, Xu H Z, Wang Z A, *et al.* 2024 *Chin. Phys. B* **33** 050301
[20] Ghosh I 2021 *Reson* **26** 671
[21] Ozols M, Roetteler M and Roland J 2013 *ACM Transactions on Computation Theory* **5** 1
[22] Moreira C and Wichert A 2016 *Frontiers in Psychology* **7** 11
[23] Moreira C and Wichert A 2018 *Journal of Mathematical Psychology* **82** 73
[24] Borujeni S E, Nannapaneni S, Nguyen N H, *et al.* 2021 *Expert Systems with Applications* **176** 114768
[25] Nguyen N and Chen K 2022 *IEEE Access* **10** 54110
[26] Harikrishnakumar R and Nannapaneni S 2023 *Expert Systems with Applications* **221** 119749
[27] Carrascal G, Botella G, Barrio A D, *et al.* 2023 *EPJ Quantum Technology* **10** 13
[28] Fathallah W, Amor N B and Leray P 2024 *International Journal of Approximate Reasoning* **175** 109307
[29] Nayak P and Seshadri K 2023 *Big Data and Artificial Intelligence: 11th International Conference*, December 7–9, 2023, Delhi, India, p. 135
[30] Oliveira M D and Barbosa L S 2023 *Foundations of Science* **28** 21
[31] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 *New J. Phys.* **18** 023023
[32] Peruzzo A, McClean J, Shadbolt P, *et al.* 2014 *Nat. Commun.* **5** 4213
[33] Chen B L and Zhang D B 2023 *Chin. Phys. Lett.* **40** 010303
[34] Mastroianni C, Plastina F, Settino J and Vinci A 2024 *IEEE Transactions on Quantum Engineering* **5** 1
[35] Zheng Q, Zhu P, Wu C, *et al.* 2024 *Science China Information Sciences* **67** 192502
[36] Pearl J 1988 *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (San Francisco, CA, United States: Morgan Kaufmann Publishers Inc.)
[37] Gyftodimos E and Flach P A 2002 *Proceedings of the ICML-2002 Workshop on Development of Representations*, July 2002, Sydney, Australia, p. 23
[38] Poole D and Mackworth A K 2010 *Artificial Intelligence: Foundations of Computational Agents* (Cambridge, UK: Cambridge University Press)
[39] Lucas A 2014 *Frontiers in Physics* **2** 5
[40] Norsys Software Corp. 2024 *Netica Application*. Available at https://norsys.com/netica.html (Accessed: 25 June 2025)