Unified Framework for Implementing Inaccurate Knowledge in Quantum Symbolic Artificial Intelligence Models

Eduardo Mosqueira-Rey[a], Samuel Magaz-Romero[b] and Vicente Moret-Bonillo[c]
Department of Computer Science and Information Technologies,
University of Coruña (CITIC), Campus de Elviña s/n, A Coruña, Spain
{*eduardo, s.magazr, vicente.moret*}*@udc.es*

Keywords: Quantum Symbolic AI, Quantum Inaccurate Knowledge, Certainty Factors, Bayesian Networks, Fuzzy Models.

Abstract: Symbolic models of Artificial Intelligence are based on defining declarative knowledge that is connected through procedural knowledge forming symbolic graphs through which reasoning flows. Both declarative and procedural knowledge can be inaccurate, which has led to the definition of different models to represent this inaccuracy. Since the functioning of quantum computers is inherently probabilistic, it has been proposed to take advantage of this nature to implement inaccurate knowledge more effectively. In this paper, we present different models for implementing inaccurate knowledge in quantum computers and propose a unified framework to represent and implement the common features of all of them.

1 INTRODUCTION

Symbolic AI builds computational models of intelligent behavior, focusing on a symbolic representation of the world and then using logic and search to solve problems. These AI models are composed of **declarative knowledge**, a set of facts that describe the real world and **procedural knowledge** that specifies how different elements of declarative knowledge relate.

The reasoning in these symbolic models is built by establishing **knowledge graphs** that are formed by declarative knowledge (the nodes) connected through procedural knowledge (the connections between nodes). These knowledge graphs are then implemented as logic rules or, more popularly, as **rule-based systems (RBS)**.

One of the problems that arises when working with symbolic AI models is that knowledge in the real world rarely is completely accurate.

In this paper, we assume that **inaccuracy** can be present in two different ways: (1) **Imprecision** when it is associated with declarative knowledge, i.e. how accurate is the description of a given fact. (2) **Uncertainty** when it is associated with procedural knowledge, i.e. the uncertainty related to the evidential strength of the causal relationship.

Over the past years, there have been several proposals to solve the issue of reasoning under inaccuracy with Quantum Computing (QC). One of the reasons behind these approaches is the probabilistic nature of QC, which aligns quite seamlessly with the preexisting models for reasoning with inaccuracy. However, there are certain drawbacks to developing practical applications with QC because nowadays quantum computations are very sensitive to noise in the currently available hardware.

In this paper, we present a brief state-of-the-art of how inaccurate knowledge can be represented in a quantum computer. We extract the common characteristics of these inaccuracy models and their corresponding quantum implementations to build a unified software framework that allows for easy implementation of a knowledge graph with inaccurate knowledge in a quantum circuit. We also carry on a study on how these models are affected by noise, given its relevance to the current viability of QC applications.

The paper is structured as follows: section 2 briefly describes classical inaccurate knowledge models and presents quantum implementations that have been proposed for these models, section 3 proposes to build a unified framework for dealing with the quantum implementation of inaccuracy, section 4 applies our unified model to a synthetic problem (*basketball*) and, finally, section 5 include the conclusions.

[a] https://orcid.org/0000-0002-4894-1067
[b] https://orcid.org/0000-0001-6438-5569
[c] https://orcid.org/0000-0002-9435-3151

Mosqueira-Rey, E., Magaz-Romero, S. and Moret-Bonillo, V.
Unified Framework for Implementing Inaccurate Knowledge in Quantum Symbolic Artificial Intelligence Models.
DOI: 10.5220/0013400200003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 1*, pages 839-846
ISBN: 978-989-758-737-5; ISSN: 2184-433X

2 QUANTUM INACCURATE KNOWLEDGE MODELS IMPLEMENTATION

In this section, we present some of the most prominent models to manage inaccurate knowledge in classical systems (Certainty Factors, Bayesian Networks, and Fuzzy Models) and we discuss hoy to to use the probabilistic nature of QC to represent the imprecision and uncertainty inherent to these models.

2.1 Certainty Factors

Proposed by (Shortliffe and Buchanan, 1975), this model was one of the first to propose a way of dealing with inaccurate knowledge. The certainty factors (CF) model is *ad hoc* in nature and therefore lacks a solid theoretical basis. However, this model was immediately accepted because of its ease of understanding and the quality of the results obtained after its application. In any case, it seems that, despite its *ad hoc* nature, probabilities are at the core of these certainty factors. In the CF model, the imprecision is represented by the certainty factors associated with facts, whereas the uncertainty is represented by certainty factors associated with rules.

We implemented CFs model in quantum computers using quantum rule-based systems, which are quantum circuits that implement knowledge in the form of production rules (Moret-Bonillo et al., 2022). We propose that inaccuracy can be obtained from a unitary matrix that operates on a specific state of the quantum system. In this context, we introduce matrix $M(\delta)$ (eq. 1), defined as follows in which the δ angle represents the inaccuracy associated with a given state (if $\delta = 0$, the statement is completely false, and when $\delta = \pi/2$, the statement is completely true):

$$M(\delta) = \begin{pmatrix} cos(\delta) & sin(\delta) \\ sin(\delta) & -cos(\delta) \end{pmatrix} \quad (1)$$

For example, suppose the following RBS with two rules: $R1 : A$ AND $B \Rightarrow X$, $R2 : X$ OR $C \Rightarrow Y$. These rules can be represented in a quantum computer as shown in Fig. 1. The circuit uses one qubit for each fact and each rule and some ancilla qubits for implementing the AND and OR operators. M gates are applied to *fact qubits* for representing imprecision and to *rule qubits* to represent uncertainty.

2.2 Bayesian Networks

A Bayesian network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph that represents the joint probability distribution over those variables. Bayesian networks reduce the space complexity by exploiting conditional dependencies in the distribution, associating with each graph node a conditional probability table for each random variable, with directed edges representing conditional dependencies. Here, the imprecision is represented by the marginal probabilities of the root nodes of the network, whereas the uncertainty is represented by the conditional probabilities that connect the non-root nodes of the network.

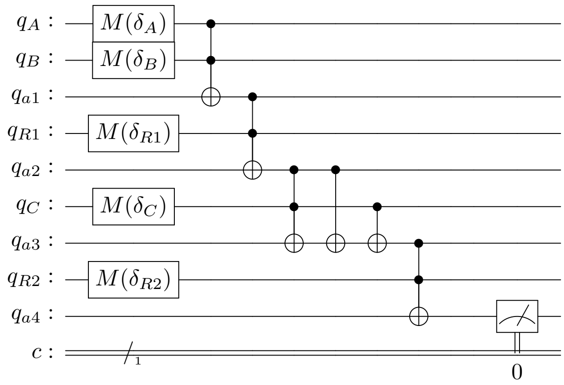

Figure 1: Quantum inferential circuit.

There are several implementations of Bayesian networks in quantum computers. Low et al. (Low et al., 2014) proposed a quantum circuit that efficiently represents the full joint distribution of a Bayesian network in which the edges in the network are mapped to conditioning nodes in the circuit. Borujeni et al. (Borujeni et al., 2021) proposed a model in which the marginal probabilities associated with root nodes (nodes without any parent nodes) are represented using rotation gates, and the conditional probability tables associated with non-root nodes are represented using controlled rotation gates. The controlled rotation gates with more than one control qubit are represented using ancilla qubits.

Let's see an example using the Titanic dataset, that lists the complete passengers and crew members on the RMS Titanic with a variable indicating whether a person did survive the sinking. Following the "*Women and children first*" rule, one can easily guess that sex and being a child have an impact on the probabilities of survival. Therefore we can represent the following Bayesian network: $(Sex) \rightarrow (Survival) \leftarrow (isChild)$

Since the nodes *Sex* and *isChild* are binary, we can represent them using a single qubit. By applying an *RY* gate with an appropriate angle, the probabilities of the root node can be mapped to the probabilities (and thus probability amplitudes) of the basis states, $|0\rangle$ and $|1\rangle$. We use a third qubit with controlled *RY* gates to represent the conditional probability tables associated with the *Survival* node (see Fig. 2).

2.3 Fuzzy Models

Fuzzy logic is a form of many-valued logic in which the truth value of the variables can be any real number between 0 and 1. The term was introduced with the 1965 proposal of fuzzy set theory by scientist Lotfi A. Zadeh (Zadeh, 1965). In these models, imprecision is represented by a *fuzzification* process in which a numeric input is assigned to a given fuzzy set with some degree of membership. Uncertainty is propagated through the system using a fuzzy implication function that triggers partially, and in parallel, several rules that include those fuzzy sets with some degree of membership different from zero. The final step is the *defuzzification* process, in which the fuzzy results are converted back into crisp results.

It is easy to see the parallelism between fuzzy models and quantum computing. The fuzzification process puts a given crisp value in a sort of "superposition" of several fuzzy variables (a person can be considered *tall* and *short* at the same time with different degrees of membership). The defuzzification process is the counterpart to the measure in quantum computers, the superposition disappears but affects how a new crisp value is obtained.

But the similarities end there, quantum mechanics is based on the Hilbert space formalism and has a probabilistic nature. On the other hand, fuzzy logic takes the concept of intersection, union, and implication of classic logic and converts them to t-norms, t-conorms, and fuzzy implications that can have several different implementations.

There are several attempts to implement fuzzy logic in a quantum computer, among the early works we can highlight (Mannucci, 2006) and (Schmitt et al., 2009), and among the more recent ones (Pykacz, 2015), (Nadaban, 2021) and (Gentili, 2021). The differences rely on how to implement the different fuzzy connectives as a composition of quantum operations (using unitary and controlled quantum gates) applied to quantum registers.

3 A UNIFIED VIEW OF QUANTUM INACCURATE KNOWLEDGE

We can see that the classical models of inaccurate reasoning present similarities: Heckerman proposed a probabilistic interpretation for Certainty Factors (Heckerman, 1986) and Gentili stated that "the terms of Bayes' formula are describable as degrees of membership to fuzzy sets, and the Bayesian inference is conceivable as a fuzzy inference" (Gentili, 2021).

This allows us to suggest the possibility of establishing a unified framework of these models to implement them in QC. In Table 1 we summarize the similarities of the models when dealing with inaccuracy and implementing them in a quantum computer.

The idea is to define of a knowledge graph and then facilitate different of this graph as different inaccuracy models on a quantum computer. Our first step was the development of software (Magaz-Romero et al., 2023) for implementing quantum rule-based systems (QRBS) defining a classical RBS.

3.1 QRBS Software

We present a software library for the implementation of QRBSs whose source code is available at (Consortium, 2024) under an open-source license.

This software library is structured in two packages, one for encoding knowledge and one for managing the quantum tasks (Moret-Bonillo et al., 2024).

The first package contains all the required elements for modeling RBSs, as illustrated in Fig. 3. In this package we can find the following classes:

- **Fact:** the building block of RBSs, as it is used to encode the declarative knowledge of the system.
- {**Not,And,Or**}**Operator:** these classes allow for the composition of the system's facts, each establishing a different relationship among them.
- **Rule:** encodes the procedural knowledge of the system, as it relates a composition of facts (the precedent) with a single fact (the consequent).
- **KnowledgeIsland:** this class aggregates different sets of rules. The rules of a knowledge island are those that aim at the same consequent.

The second package contains all the elements to define a QRBS, as well as the tools to evaluate and execute it. These components (Fig. 4) are:

- **WorkingMemory, InferenceEngine:** store the declarative and procedural knowledge.
- **QRBS:** used to initialize the system and populate it with the corresponding knowledge elements.
- **QPU:** interface defining how the QRBSs are operated, with regards to evaluation and execution.

The library provides the following functionalities:

- **RBS modeling:** users can model a rule-based system by encoding declarative and procedural knowledge into a system, establishing different

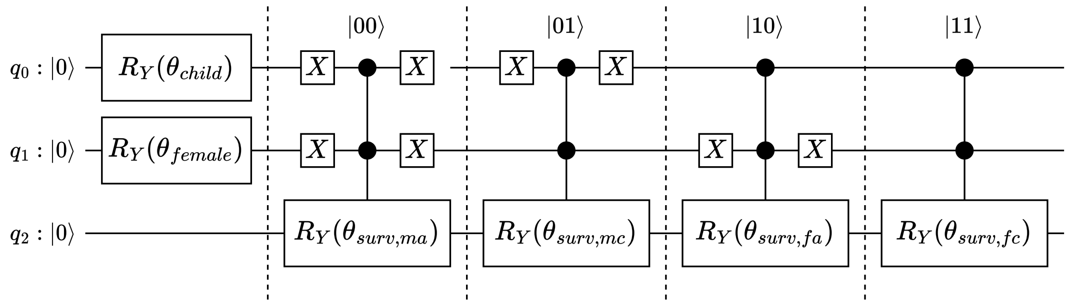

Figure 2: Quantum Bayesian network circuit.

Table 1: Summary on representing the imprecision and uncertainty of different models in classical and quantum computers.

	Classical		Quantum	
Model	Imprecision	Uncertainty	Imprecision	Uncertainty
Certainty Factors	CFs in facts	CFs in rules	M gate in facts	M gate in rules
Bayesian Networks	Marginal probabilities	Conditional probabilities	RY gates in root nodes	CRY gates in non-root nodes
Fuzzy models	Fuzzification of facts	Fuzzy implication	Superposition of quantum states	Controlled quantum gates

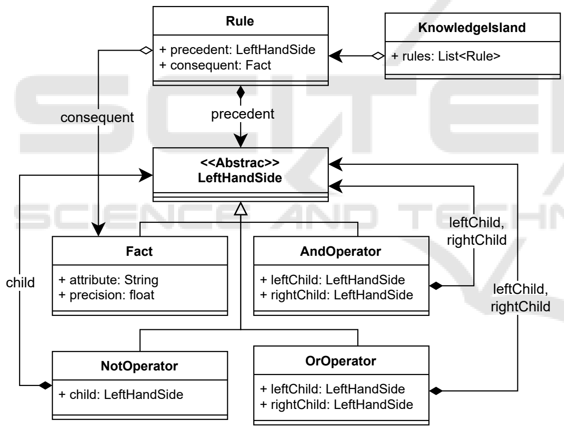

Figure 3: Class diagram of the knowledge rep. package.

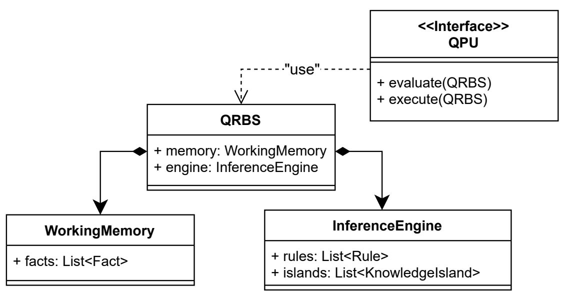

Figure 4: Class diagram of the QRBS package

facts with their respecting imprecision, and relating them through rules with uncertainty.

- **Automated QRBS implementation in different models:** after defining an RBS, users can automatically obtain the corresponding QRBS in the different models available without providing any additional information.
- **Evaluate and execute QRBS:** users can evaluate their QRBS to analyze their viability , and if favorable execute it to obtain the corresponding results. These are encoded as the certainty values of the rules' consequents, which users can consult.

The general QRBS workflow would be as follows:

1. The user defines the elements of a classical rule-based system (facts, rules, etc.)
2. The user establishes the level of inaccuracy in the elements of the system.
3. An implementation of the QPU interface uses this information to build the corresponding quantum routine for a given quantum computer.
4. This implementation is executed, yielding the results for the consequents of the different rules.
5. The values of the consequents are updated so the user can consult them.

With this library, we allow for an easy management of the knowledge elements of RBSs, independently from a specific model of uncertainty. Once the declarative and procedural knowledge of the system is identified, it is trivial to encode it with the library.

Furthermore, it enables a fast workflow for experimenting with these systems. Once its structure has been defined, the user can easily tweak small aspects directly (e.g., the accuracy values of each of the elements) and automatically obtain the corresponding quantum routines for each of the models available.

3.2 On the Subject of Noisy Quantum Computations

Quantum Computing has gained enormous momentum in recent years thanks to the development of the first publicly available quantum computers, although these machines are still far from being comparable with classical computers. While classical computers have enough capacity for error correction , for quantum computers it is a luxury (Wolfowicz et al., 2021).

Nevertheless, current quantum computers are worth using and researching . This is where NISQ algorithms come into play, as they are designed to work under these circumstances. Their goal is to leverage the available resources to perform classically challenging tasks (Bharti et al., 2022).

We have developed our approach within this context. While the quantum routines that result from the different models are themselves limited by the number of qubits, the framework itself is ready to scale when quantum computers with more capacity are made available. However, we must test the framework's robustness against noise.

4 SYNTHETIC CASE: BASKETBALL PROBLEM

To illustrate the framework and the use of the QRBS software, we propose to study the following case, which we name the basketball problem.

4.1 Description of the Problem

This problem is based on a simulated case, where a basketball coach has to make a new team from the players available. He decides to compare them by evaluating two statistics: the scored shots out of 20 as a measurement of skill, and the height of the player.

Each stat is split into five categories: *Worst*, *Bad*, *Regular*, *Good* and *Best*. Therefore, we have 10 input facts for the system, with a level of inaccuracy assigned depending on the input of the stat.

For the evaluation of the player, we consider four categories: *Bad* (which is discarded directly), *Regular*, *Good*, and *Excellent*. In this case, we obtain the precision levels for each category from the inference process of executing the system.

Once we have defined the facts of the system, we define the rules that relate the stats to the evaluations. We consider those in Table 2.

To evaluate the system, we generate a batch of synthetic players, i.e. a pair of shots/height, and input this data into the system to analyze the results. Table 3 shows the generated data for the synthetic players. We have implemented the problem classically to obtain an expected value for each synthetic player, so we can later validate the output.

For each of these players, we initialize the precision of the facts according to their pair of statistics. For the sake of briefness, we assume that the certainty for the rules is always 1.00.

We have defined this set of players with values spread across the range of evaluation so we can later evaluate the effect of noise in them.

4.2 Implementation of the Problem

We test the three models available in the QRBS software,as each should yield similar results to the other two.As stated in Table 1, each model has a different quantum implementation. To illustrate them, we present the quantum routines corresponding for Rule 5 (see Table 2). The same procedure is applied for the rest of the rules to compose the corresponding entire quantum circuit.

For Certainty Factors, the quantum routine is the one shown in Fig. 5. $M(\theta)$ is used to initialize the facts' qubits for their imprecision and the rules' qubits for their uncertainty. We apply the quantum operators according to the logical associations between the facts, as well as the ones for the inference of the rules.

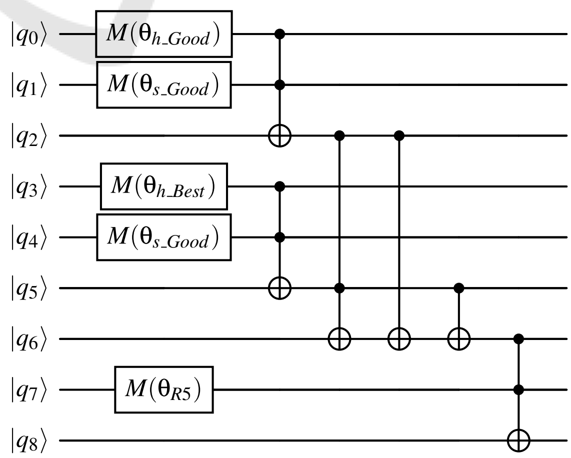

Figure 5: Quantum routine for Rule 5 with CF model.

For Bayesian networks, we obtain the quantum routine from Fig. 6. We use $RY(\theta)$ for the declara-

Table 2: Rules for the basketball problem.

ID	Precedent	Output
1	IF shots is *Worst* OR shots is *Bad*	*Bad*
2	IF height is *Worst* OR height is *Bad*	*Bad*
3	IF height is *Regular* OR shots is *Regular*	*Bad*
4	IF (height is *Regular* AND shots is *Good*) OR (height is *Regular* AND shots is *Best*) OR (height is *Good* AND shots is *Regular*) OR (height is *Best* AND shots is *Regular*)	*Regular*
5	IF (height is *Good* AND shots is *Good*) OR (height is *Best* AND shots is *Good*)	*Good*
6	IF (height is *Good* AND shots is *Best*) OR (height is *Best* AND shots is *Best*)	*Excellent*

Table 3: Generated data for the basketball problem.

ID	Scored shots	Height	Expected value
P1	15	203	50.00
P2	16	198	64.60
P3	17	188	16.90
P4	17	193	56.10
P5	18	176	0.00
P6	18	186	17.60
P7	18	200	83.70

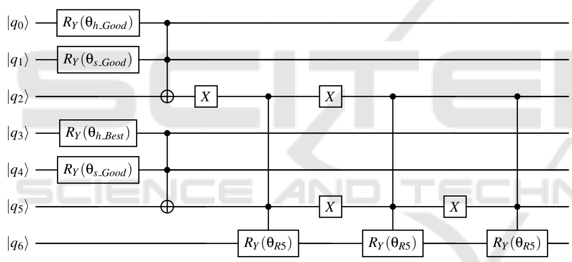

Figure 6: Quantum routine for Rule 5 with Bayes model.

tive knowledge (root nodes), and $CRY(\theta)$ for the procedural knowledge (non-root nodes). We apply the quantum operators to model the relationship between nodes, following the rules of the system.

For Fuzzy models, the resulting quantum routine is illustrated in Fig. 7. For this model, we apply $R_Y(\theta)$ gates for the inaccuracy of both the facts and the rules. We implement the *AND* with a *CCNOT* and the *OR* with a composition of *X* and *CCNOT*.

With these quantum routines, we show how each of the models is implemented with slight differences. It is worth mentioning that the routines have been obtained from the same classical inferential circuit.

Regarding the noise models, we follow the error model based on thermal relaxation with the qubit environment. Table 4 illustrates the parameters applied for each noise model, as we introduce noise in $U1$, $U2$, and $U3$ quantum gates, in the CX controlled operator, and in the measuring process. The first model is noiseless and validates the values from the quantum

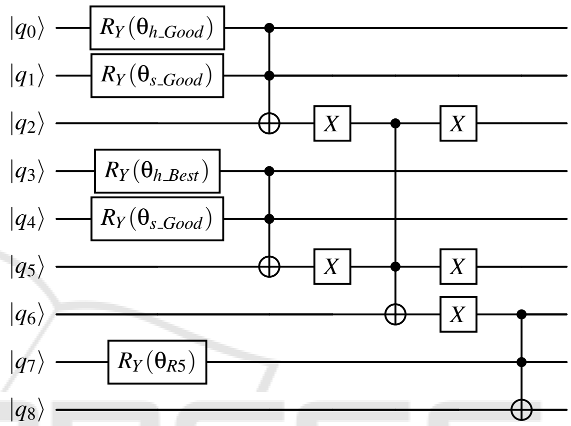

Figure 7: Quantum routine for Rule 5 with Fuzzy model.

Table 4: Parameters (in nanoseconds) for the noise models.

Noise	U1	U2	U3	CX	Measure
0	*0*	*0*	*0*	*0*	*0*
1	71	142	428	1428	1428
2	142	285	857	2857	2857
3	214	428	1285	4285	4285
4	285	571	1714	5714	5714
5	357	714	2142	7142	7142
6	428	857	2571	8571	8571
7	500	1000	3000	10000	10000

routines against the classical expected values.

4.3 Results

In the first place, we must validate that the quantum routines' outputs match with the expected ones. For that, we evaluate the quantum routine of each model with the noiseless model, shown in Fig. 8.

It can be observed how the obtained values correspond to the expected values from the classical implementation, therefore validating our framework. As expected, the results obtained for each model are similar to the ones from the other two, with some of them even being identical.

We continue with the results of the experiments with the different noise levels, shown in Fig. 9.

These results showcase how noise affects the values returned by the quantum routines. The output value tends to 50 as the noise becomes more aggressive. As more noise is introduced in a reasoning process, the conclusions provide less information, which is reflected as an evaluation of 50: the player is equally “good” and “bad”.

However, for the proposed framework, the influence of noise does not affect the problem’s goal. Since noise affects all cases equally, the players’ order is preserved. There are some points where this proportion does not hold up as strongly as one would prefer, due to the probabilistic nature of QC.

We have calculated the error for each noise level output for the noiseless model, which can be seen in Fig. 10. In general, the CF model presents the highest error value, yet it could be argued that also preserves the proportion of the players’ values the best.

In general, the models developed in this work present enough robustness against the noise models that have been experimented with. This makes them appropriate to be used during the NISQ era of QC.

5 CONCLUSIONS

AI has classically been divided into two worlds, the symbolic world and the connectionist world. Today we can say that the latter has surpassed the former in terms of results. However, many researchers believe that an exclusively connectionist model is not enough to achieve Artificial General Intelligence (AGI), i.e. human-level AI, and that a mixed approach is needed.

Similarly, quantum AI has tended to machine learning (ML) developments because such models deal with massive parallel processing of high-dimensional data, it seems that quantum computers can be an advantage when designing ML algorithms.

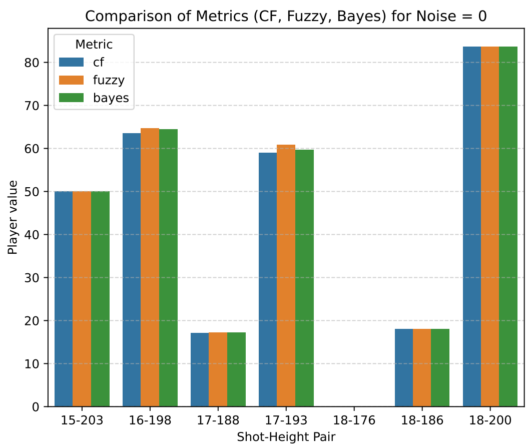

Figure 8: Scores for the synthetic players without noise.

However, we think that there is room in the quantum world for symbolic AI. Since QC is inherently probabilistic, we can take advantage of this nature to implement inaccurate knowledge more effectively.

This idea has taken form in the QRBS software presented here. With this framework, users can model their knowledge classically and implement it automatically in a given quantum computer.

On top of that, the results obtained prove both that the framework presented is valid for the implementation of symbolic AI models in a QC, and that it manages inaccurate knowledge successfully. We also have demonstrated how our framework is suitable for NISQ quantum computers, making it viable before noise-correcting quantum computers are available.

This development presents a first step towards a quantum symbolic AI library that can implement generically classical symbolic AI models in a quantum computer, and that allows us to take advantage of the probabilistic nature of quantum computers to find a quantum advantage in such implementation.

ACKNOWLEDGEMENTS

This work has been supported by the EU’s Horizon 2020 under project NEASQC (grant No 951821), the State Research Agency of the Spanish Government (Grant PID2023-147422OB-I00) and by the Xunta de Galicia (Grant ED431C 2022/44), supported by the EU European Regional Development Fund (ERDF). CITIC, as a center accredited for excellence within the Galician University System and a member of the CIGUS Network, receives subsidies from the Department of Education, Science, Universities, and Vocational Training of the Xunta de Galicia. Additionally, it is co-financed by the EU through the FEDER Galicia 2021-27 operational program (Ref. ED431G 2023/01). We thank the support from Ministry for Digital Transformation and Civil Service and Next-GenerationEU/RRF (TSI-100925-2023-1). SMR has received funding from Xunta de Galicia (grant ED481A 2023/008).

REFERENCES

Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J. S., Menke, T., Mok, W.-K., Sim, S., Kwek, L.-C., and Aspuru-Guzik, A. (2022). Noisy

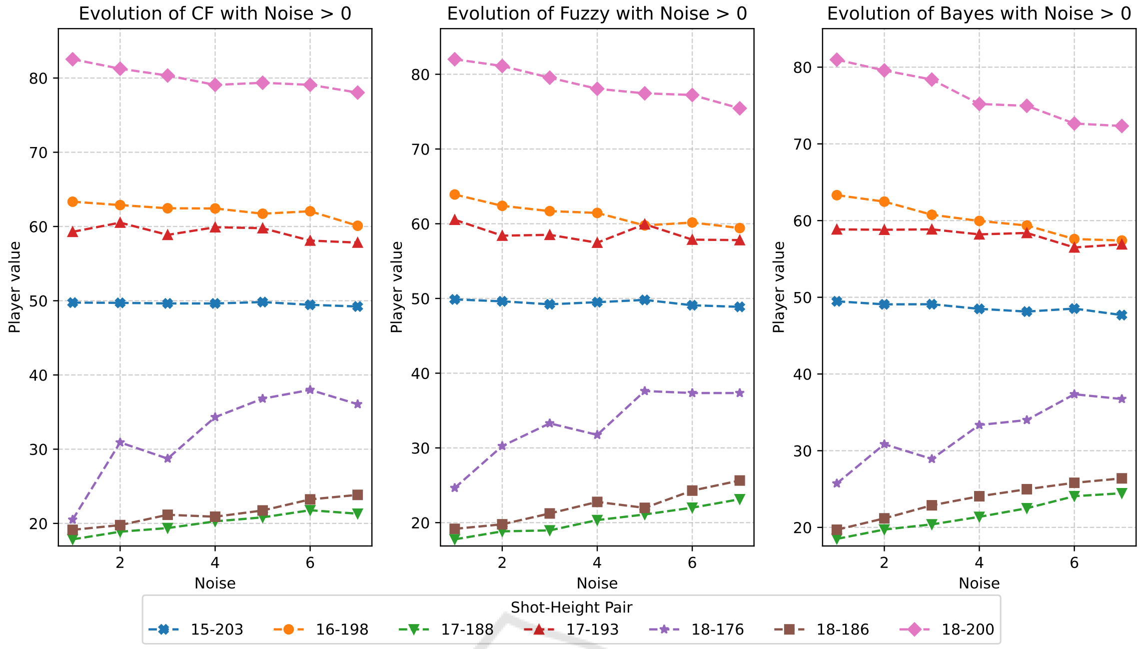

Figure 9: Evolution of the CF, Fuzzy, and Bayes models regarding the noise levels.

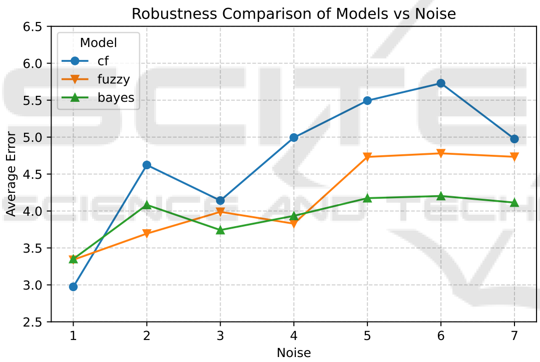

Figure 10: Robustness of the model for each noise level.

intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1).

Borujeni, S. E., Nannapaneni, S., Nguyen, N. H., Behrman, E. C., and Steck, J. E. (2021). Quantum circuit representation of bayesian networks. *Expert Systems with Applications*, 176:114768.

Consortium, N. (2024). Quantum rule-based system (qrbs) library.

Gentili, P. L. (2021). Establishing a new link between fuzzy logic, neuroscience, and quantum mechanics through bayesian probability: Perspectives in artificial intelligence and unconventional computing. *Molecules*, 26(19).

Heckerman, D. (1986). Probabilistic interpretations for mycin's certainty factors. In Kanal, L. N. and Lemmer, J. F., editors, *Uncertainty in Artificial Intelligence*, volume 4 of *Machine Intelligence and Pattern Recognition*, pages 167–196. North-Holland.

Low, G. H., Yoder, T. J., and Chuang, I. L. (2014). Quantum inference on bayesian networks. *Phys. Rev. A*, 89:062315.

Magaz-Romero, S., Mosqueira-Rey, E., Álvarez-Estévez, D., and Moret-Bonillo, V. (2023). Quantum factory method: A software engineering approach to deal with incompatibilities in quantum libraries. In *Int. Conf. on Computational Science (ICCS-2023)*. in press.

Mannucci, M. (2006). Quantum fuzzy sets: Blending fuzzy set theory and quantum computation. *CoRR*, abs/cs/0604064.

Moret-Bonillo, V., Magaz-Romero, S., and Mosqueira-Rey, E. (2022). Quantum computing for dealing with inaccurate knowledge related to the certainty factors model. *Mathematics*, 10(2).

Moret-Bonillo, V., Magaz-Romero, S., Mosqueira-Rey, E., and Alvarez-Estévez, D. (2024). D6.14 Final QRBS software and IDC application.

Nadaban, S. (2021). From classical logic to fuzzy logic and quantum logic: A general view. *International Journal of Computers communications & control*, 16(1).

Pykacz, J. (2015). *Quantum physics, fuzzy sets and logic*. Springer.

Schmitt, I., Nürnberger, A., and Lehrack, S. (2009). On the relation between fuzzy and quantum logic. In Seising, R., editor, *Views on Fuzzy Sets and Systems from Different Perspectives*, pages 417–438. Springer.

Shortliffe, E. H. and Buchanan, B. G. (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3):351–379.

Wolfowicz, G., Heremans, F. J., Anderson, C. P., Kanai, S., Seo, H., Gali, A., Galli, G., and Awschalom, D. D. (2021). Quantum guidelines for solid-state spin defects. *Nature Reviews Materials*, 6(10):906–925.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.