Integrating Quantum CI and Generative AI for Taiwanese/English Co-Learning

Chang-Shing Lee[1], Mei-Hui Wang[1], Chih-Yu Chen[1], Sheng-Chi Yang[1]
Marek Reformat[2,3], Naoyuki Kubota[4], Amir Pourabdollah[5]
[1]Department of Computer Science and Information Engineering, National University of Tainan, Taiwan
[2]Department of Electrical and Computer Engineering, University of Alberta, Canada
[3]Information Technology Institute, University of Social Sciences, Poland
[4]Department of Mechanical Systems Engineering, Tokyo Metropolitan University, Japan
[5]School of Science and Technology, Nottingham Trent University, Nottingham, UK
email: leecs@mail.nutn.edu.tw

Abstract

This paper proposes a *Quantum Computational Intelligence* (QCI) model integrated with *Generative Artificial Intelligence* (GAI) for Taiwanese/English language co-learning applications within human-machine interactions, focusing on *Trustworthy AI Dialogue Engine* (TAIDE)-based knowledge graph construction and multimodal data transformation. The QCI model comprises two main phases: human-machine interaction and data processing for *quantum circuit* generation and real-world applications. During the human-machine interaction phase, a synergy between *Human Intelligence* (HI) and *Machine Intelligence* (MI) enables young students to gain familiarity with CI that converges with QCI. The second phase involves data processing, which encompasses stages of data preprocessing, analysis, and evaluation. The methodology is applied to two distinct applications: ***Application 1*** focuses on constructing a *knowledge graph* using the *Ollama* platform and the TAIDE model developed by the Taiwanese government based on the LLaMa 2 model. ***Application 2*** addresses the GAI images to text/voice, and text/voice to GAI images, depending on the type of Taiwanese/English data collected. Subsequently, the QCI model is refined through Particle Swarm Optimization (PSO) and Genetic Algorithm Neural Networks (GANN). Moreover, a Quantum Fuzzy Inference Mechanism (QFIM) is integrated into the developed *QCI&AI-FML learning platform* to generate *quantum circuit*s for the QCI model, which helps teach young students and facilitate their learning of QCI. The experimental results indicate that the QCI model significantly enhances human-machine collaboration. Looking forward, we plan to extend the QCI model to reach more young learners.

Keywords: Human-Machine Interaction, Quantum Computational Intelligence, Generative AI, Particle Swarm Optimization, Genetic Algorithm Neural Network, TAIDE, Knowledge Graph

1. Introduction

The convergence of artificial intelligence, robotics, and communication technologies in research collaborations manifests its most compelling application in 'edutainment'—a blend of education and entertainment (Lee et al., 2019). This includes methods such as learning on, by, and with robots. The usability of e-learning systems has recently seen dramatic improvements, largely due to advancements in *Generative Artificial Intelligence* (GAI). As a result, knowledge exchange through cross-disciplinary, cross-cultural, and cross-generational communication has become much more accessible. Moreover, these technologies allow us to build systems that embody our ideas through a process of trial and error, fostering the growth of creativity. A fundamental component of creativity is inspiration, which often arises from the 'small wonders' we encounter in everyday life. While these moments might seem trivial to some, they can present significant challenges or opportunities to others. Thus, there is a compelling case for shifting from data-driven AI to inspiration-driven AI, emphasizing the importance of nuanced human experiences and perspectives in the development of intelligent systems.

The emergence of GAI models, such as ChatGPT (OpenAI, 2024), has garnered significant public interest. The rapid development of these models has sparked intense debates regarding their advantages, limitations, and potential risks (Baldassarre et al., 2024). GAI has notably advanced the field of text-to-image and image-to-text generation (Wang et al., 2024; Lee et al., 2024a). Models like Stable Diffusion (Rombach et al., 2022) and DALL-E (OpenAI, 2024) have demonstrated remarkable capabilities in creating high-quality images. As a result, GAI models are being applied in diverse areas, including visual art creation, news illustration, industrial design, healthcare, finance, and education (Baldassarre et al., 2024; Wang et al., 2024; Lee et al., 2024a).

Quantum Computational Intelligence (QCI) is a field actively pursued by researchers and developers for its practical applications and solutions to real-life problems. Quantum computers, which utilize quantum bits (qubits) for information storage and processing, offer a unique capability compared to traditional computers. Unlike traditional computers that operate using binary digits (0 and 1), quantum computers can process 0 and 1 simultaneously. This ability enables quantum computers to rapidly solve specific computational problems that are intractable for conventional computers, thereby enhancing computational performance. Consequently, quantum computation employs complex Hilbert space as its mathematical foundation, rather than traditional Boolean Algebra. One significant application of *Quantum Computing* (QC) is to accelerate the typical inference processes of human reasoning (Acampora, 2024). This advancement facilitates the development of control or decision support systems capable of efficiently handling large datasets in an interpretable manner. Acampora et al. (2023) proposed a *Quantum Fuzzy Inference Engine* (QFIE) to leverage the advantages of quantum computing in enhancing traditional fuzzy inference systems and rules. Acampora and Vitiello (2023) integrated IEEE Standard 1855 (IEEE CIS, 2016) and QFIE to provide the fuzzy community with controllers transparent to hardware and efficient in terms of fuzzy rule execution. QFIE is especially crucial as the number of rules in fuzzy systems can grow exponentially with the addition of input variables, causing classical inference algorithms to spend exponential time evaluating fuzzy rules. However, with QFIE, rules can be evaluated on a quantum computer in polynomial time, achieving a quantum advantage in fuzzy inference processes. By enabling simultaneous execution and accelerated processing, QFIE aims to improve performance and efficiency in distributed or big data environments, significantly boosting computational efficiency.

Computational Intelligence (CI) is considered one of the most promising and yet-to-be-fully-explored fields (Ishibuchi, 2019). To promote STEM (Science, Technology, Engineering, and Mathematics) education at the pre-university level, we have collaborated with IEEE CIS and IEEE Region 10 (R10) to organize activities that extend outreach to undergraduate and pre-university students, providing an educational experience through engaging competitions (OASE Lab., 2019). The event introduces QCI and GAI techniques and their learning models to students, motivating them to study CI techniques and experience QCI learning tools with hardware/software and their applications (Lee et al. 2022; 2023a, Lee et al. 2024a). Additionally, leading researchers in CI and IEEE volunteers conduct plenary lectures and workshops. The associated competition with the event is expected to draw more attention from scholars worldwide to IEEE-related conferences and initiate collaborations with companies interested in supporting young scholars. We hope that early engagement with STEM education will inspire students to become interested in QCI and GAI learning, further encouraging young students and researchers to apply CI techniques and methodologies in real-world scenarios.

This paper applied the proposed method to two primary applications: 1) ***Application 1***: Utilize the *Ollama* platform and the TAIDE model, a *Trustworthy AI Dialogue Engine* (TAIDE) developed by the Taiwanese government based on the LLaMa 2 model (NARLabs, 2024), to construct the *knowledge graph* after extracting the important *concepts* and *relations* among them from the data collection. 2) ***Application 2***: Employ the *Multimodal GAI Interactive Learning (GAIIL) platform* for multi-modal co-learning in Taiwanese or English with machines, accommodating input types such as text and speech, and using evolutionary computation methods, including *Particle Swarm Optimization* (PSO) and *Genetic Algorithm Neural Networks* (GANN), for CI model optimization. A *Quantum Fuzzy Inference Mechanism* (QFIM) based on QFIE is integrated to infer the results and provide a *quantum circuit* to the QCI model. Experimental results indicate that the QCI model enhanced with GAI images significantly improves human-machine collaboration and serves as an effective educational tool for high school and undergraduate students to understand QCI and GAI *concepts*.

The remainder of this paper is structured as follows: Section 2 describes the structure of the QCI model for human-machine interaction, the activities collaborated with IEEE CIS and IEEE R10, and *QCI learning tools* with hardware/software. Section 3 introduces GAI applications on the *knowledge graph*, specifically extracting *concepts* from the data collection and then creating a *knowledge graph*. The QCI model with GAI for *Taiwanese/English Co-Learning* based on text/voice and image generation applications is proposed in Section 4. Experimental results are presented in Section 5, and the conclusions are drawn in Section 6.

2. Quantum Computational Intelligence Model for Human-Machine Interaction

This section describes the QCI model for human-machine interaction. The structure of human-machine interaction is introduced in Section 2.1, the *QCI sandbox for pre-university and undergraduate students* is presented in Section 2.2, and the connection between the *QCI&GAI learning platform with learning tools*, along with their introduction, is described in Sections 2.3 and 2.4, respectively.

2.1 Quantum CI Model Applications for Human-Machine Interaction Structure

The *Quantum CI* (QCI) model for human-machine interaction is structured based on the principles of co-learning in QCI and GAI, involving both human and machine participants. The model comprises two phases: 1) *human and machine interaction* and 2) *data processing* for *quantum circuit* generation and real-world applications, as shown in Fig. 1 and described in the following sections:

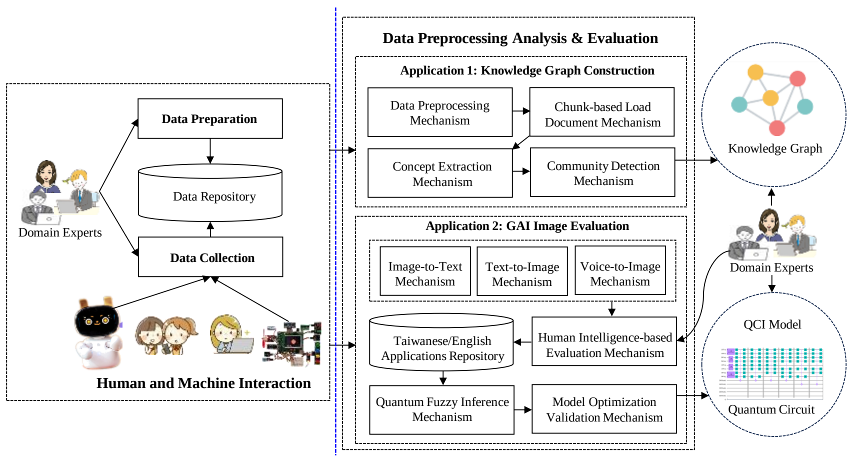


Fig. 1. Quantum CI model applications for human-machine interaction structure.

- **Human and Machine Interaction:** 1) The interaction between humans and machines is centered on a co-learning process merging *Human Intelligence* (HI) with *Machine Intelligence* (MI). This process includes various learning types, such as *concept-based learning*, *experience-based learning*, *operation-based learning*, *practice-based learning*, *application-based learning*, and *expression-based learning*, each corresponding to different educational levels from elementary school to university, providing diverse learning experiences and practical applications for various age groups. 2) CI acts as a bridge connecting HI and MI, enhancing the collaborative nature of the learning model. The co-learning process takes place at this intersection, combining the strengths of human and machine learning methods. The HI component of the model emphasizes the learning and application cycle, focusing on *Neural Networks* (NNs) for *perception* and *Fuzzy Logic* (FL) for *cognition*. 4) These processes highlight the importance of understanding human learning to synergize with machine intelligence effectively. It also underscores the significance of QCI for integrating with FL and incorporating *Evolutionary Computation* (EC) in the *evolutionary* aspect of machine learning.

- **Data Processing:** The data processing phase of a model designed for human-machine interaction is divided into two main applications. 1) ***Application 1*** comprises four mechanisms that construct a *knowledge graph*, which visually represents extracted *concepts* and their *interrelations* from collected texts or PDF files. 2) ***Application 2*** focuses on GAI performance, encompassing the conversion of images to text/voice, text descriptions to images, and Taiwanese/English spoken words to visual representations. After data collection, we assess the performance of generative texts and images based on HI criteria. Domain experts are instrumental in this phase, lending their expertise to refine and validate the model's performance.
- The QCI model, which integrates QCI knowledge and inference models through quantum fuzzy inference, is presented as a *quantum circuit* for real-world applications. The QFIE is used in the co-learning framework, where users make quantum fuzzy inferences for their specific CI model through the QFIE integrated into the developed *QCI&AI-FML learning platform*. This demonstrates the fusion of quantum computing principles with CI in the context of human-machine interaction.

2.2 Quantum Computational Intelligence for Pre-University and Undergraduate Students Sandbox

Fig. 2 illustrates the growth in human-machine relations from 2009 to 2024, marking a transition from competition to co-learning paradigms. The events organized by IEEE groups such as the IEEE *Computational Intelligence Society* (CIS), IEEE Region 10 *Strategic Planning and New Initiatives Committee* (SPNIC), and the IEEE *Systems, Man, and Cybernetics Society* (IEEE SMC) highlight the milestones of this evolution. The world map pinpoints the locations and dates of significant events that illustrate this progression.

The period between 2009 and 2016, known as the *Human vs. Computer Go Competition* era, represented an initial stage where HI contended with MI under handicapped conditions in the realm of strategic board games such as the game of Go (Lee et al., 2016). Conversely, in the span from 2017 to 2024, *Human and*

Machine Co-Learning indicates a shift away from competition towards a partnership and shared learning between humans and machines (Lee et al., 2020). This transition also introduces diverse types of learning, enriching the educational experiences across different age groups and enabling young learners to become acquainted with CI as it integrates with QCI (Lee et al., 2023b; Lee et al., 2024a). For detailed information, please refer to Item No. 1 in the Appendix.

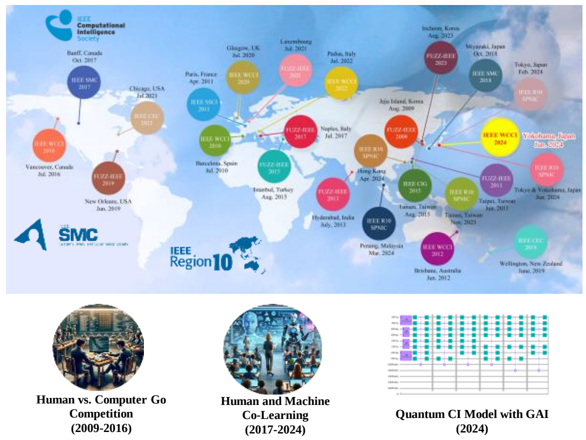


Fig. 2. Growth and development in the relationship between humans and machines at IEEE CIS and IEEE SMC.

2.3 Connection between QCI&GAI Learning Platform and Learning Tools

Fig. 3 depicts a human and machine co-learning setup utilizing the *Message Queuing Telemetry Transport* (MQTT) protocol. Learners access the *QCI&AI-FML learning platform* to establish connections with various MQTT brokers, including NUWA Robotics, Webduino, and NUTN, or they may select a custom service. This setup requires entering relevant details such as username, password, address, and topics for subscription and publishing. Learners interact with both the *learning platform* and the *QCI&AI robot* and *learning tool* using laptops or tablets, facilitating the exchange of messages between clients and the MQTT broker.

Once configured, the co-learning between the *CI knowledge model* based on HI as well as the *machine model* based on MI can be successfully established. When the machine receives data, it responds with actions such as displaying information on its face, verbalizing the information, and triggering hardware actions. Conversely, the machine can also automatically collect data to publish to the learning platform for inference and then receive the inferred result to respond with actions, achieving a two-way connection. The

QCI&AI-FML learning platform supports *CI inference* and *QCI inference*. Learners first upload their CI model to the platform and choose *CI inference* or *QCI inference*; however, *QCI Inference* only supports the *Mamdani inference model*. During *QCI inference*, the *QCI inference model* displays membership degrees for linguistic terms and the probability of each possible consequence in learners' real-world application system. The final defuzzified result is derived from the aggregated probabilities.

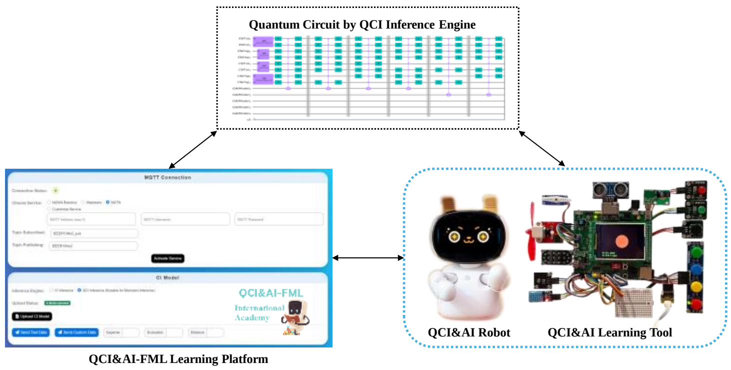


Fig. 3. Connection between *QCI&GAI Learning Platform* and *Learning Tools*.

2.4 *Introduction to QCI&AI-FML Learning Platform and QCI&AI Learning Tools*

The *QCI&AI robot*, suitable for children, interacts with its environment and learners using MQTT and *Blockly programming*. Conversely, the *QCI&AI learning tool*, designed for high school and undergraduate students, is equipped with various modules such as servos, fans, speakers, light sensors, and cameras. These modules are programmed in Python to receive commands and send data via MQTT. Through this system, learners can interact in real time with a *QCI&AI-enabled robot* and *learning tool*, experiencing an interactive educational environment that fosters collaborative learning with machines. The *QCI&AI-FML learning platform* comprises six main components, including the *CI Knowledge Model*, *CI Inference Model*, *Conduct CI Inference*, *CI Model Validation*, *CI Machine Learning*, and *QCI Model*. Table 1 provides descriptions for each component. For detailed information on the user interface, please refer to Item No. 2 in the Appendix.

Table 1. Description of *QCI&AI-FML learning platform* components.

- **CI Knowledge Model:** Enable learners to create a CI model tailored to a specific real-world application, such as recommending tourist spots. This process involves specifying the number of variables, with inputs like *Expense*, *Evaluation*, and *Distance*, and the output being the *Travel Recommendation Level*. Learners must then establish the shapes of the membership functions for these terms to complete the knowledge model for recommending tourist spots.
- **CI Inference Model:** The creation of a CI inference model begins by formulating inference rules that reflect various combinations of input variables, each defined by a set of linguistic terms. This process yields a series of fuzzy system 'if-

then' rules. For instance, one rule may state that if the *Expense is Low, the Evaluation is Bad, and the Distance is Near, then the Recommendation Level would be 'Not Recommended.'* The linguistic terms used in the output are subjective and may vary based on individual perspectives and judgments. After establishing the rules, the CI model can be loaded to complete the creation of the inference model.

- **Conduct CI Inference:** Conducting CI inference involves inputting data for variables and then inferring the *Recommendation Level*. For example, *a high Expense, a bad Evaluation, and a far Distance would lead to a low Recommendation Level*, indicating that the platform does not recommend visiting the place. Learners can load and save inference data individually or in batches, and download inference results with or without semantics.
- **CI Model Validation:** Validating expert knowledge involves creating and loading expert data onto the platform. The platform processes the data in batches and employs semantic matching accuracy, along with Mean Square Error (MSE) and Root Mean Square Error (RMSE), to evaluate its consistency with expert interpretations and assess the model's prediction error. For example, a discrepancy between the predicted value (82.381, *very recommended*) and the expert's value (70, *very recommended*) results in a squared error of 153.288. However, their semantics match, indicating that the assessment remains the same despite the numerical difference.
- **CI Machine Learning:** CI machine learning enables learners to train models for their real-world applications using the PSO method. This evolutionary computation technique is inspired by the collective behavior of animal swarms. The training process involves loading training data, setting the number of particles and iterations, and then initiating the training. Upon completion, users can download the training results and save the trained model. The platform displays the knowledge model before and after training, highlighting changes in the membership function shape to align with expert knowledge.
- **QCI Model:** The QCI Model feature allows users to save, download, and validate CI models with QCI circuits by the IEEE 1855 standard.

3. Generative AI with Ollama and TAIDE Model for Knowledge Graph Application

This section describes the GAI with the *Ollama* platform and TAIDE model for *knowledge graph* applications. The structure of the *GAI-based Knowledge Graph* (GAIKG) is introduced in Section 3.1, and the *concept extraction mechanism* and *community detection mechanism* of GAIKG are presented in Sections 3.2 and 3.3, respectively.

3.1 GAIKG with Ollama Platform and TAIDE Model for Knowledge Graph Applications

Fig. 4 illustrates the GAIKG process of transforming context into a *knowledge graph* using a *Large Language Model* (LLM), such as the TAIDE model or the *Ollama* platform. Derived from a repository of human and machine interactions, chunks of context are formulated into user prompts for the LLM.

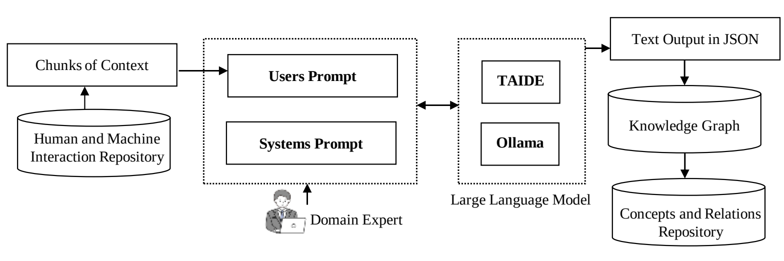


Fig. 4. GAIKG with the *Ollama* platform and TAIDE model for *knowledge graph* applications.

The system's prompt of GAIKG, as defined by domain experts, specifies that the model should operate as a network graph maker, extracting terms and their relationships from the context and formatting the output in JSON. These prompts instruct the LLMs to discern and distill key *concepts* and their *relationships* from the text. The two LLM models receive both the GAIKG system and user prompts and process the query to generate a response chunk. Ultimately, the LLM produces an output in JSON format that includes the extracted *concepts* and their *relations*. Using this JSON output, a *knowledge graph* is constructed, visually representing the *relationships* between key *concepts* identified in the context, and showcasing the transformation from raw data to structured as well as visual information. Additionally, the *knowledge graph* quantifies the size and degree of *concepts* and the weight of the *relations*, which may be leveraged for further computational processes or analysis.

3.2 Concept Extraction Mechanism of GAIKG

This paper adopts an open-source (Rahulnyk, 2024) as an initial version to create a *knowledge graph* for the documents from the repository of human-machine interaction documents. This is achieved by extracting *concepts* and *relationships* that are related in the vicinity of two concepts from each chunk of the documents using the model on the *Ollama* platform and the TAIDE model. Once the concept pairs and semantic relationships between them are extracted, this information is used to build the *knowledge graph* for analyzing the texts collected from human-machine interaction. Table 2 shows the *concept extraction mechanism* algorithm of GAIKG based on the *Ollama* platform and the TAIDE model. Captions are listed in Item No. 3 in the Appendix, resulting in a total of 19 chunks. Figs. 5 and 6 show the results of the second and the last chunks, respectively. These figures illustrate the text of each chunk and the extracted *concepts* and *relationships* using the TAIDE model. In these figures, the extracted *concepts* are marked in bold and italics.

Table 2. *Concept extraction mechanism* algorithm of GAIKG based on the *Ollama* platform and TAIDE model.

Input:
1. Documents from the human and machine interaction repository, located in the directory.
2. ***LLM TAIDE***, featuring 13 billion parameters and 4-bit quantization.
3. ***LLM Zephyr on the Ollama development platform***, featuring 7 billion parameters and 4-bit quantization.
4. *SYS_PROMPT*: '*You are a network graph maker who extracts terms and their relations from a given context. You are provided with a chunk of context. Your task is to extract the ontology of terms mentioned in the given context. These terms should represent the key concepts as per the context. Format your output as a list in JSON.*'
Output:
1. *df_KG:* Extracted concept pairs and semantic relationships between them from input documents, with the weight among two concepts.
2. *df_node:* Extracted concepts with the information of the name and size.

Method:

Step 1: Split the documents into text chunks chk_1, chk_2, …, chk_n, where $1 \le n \le N$ /*N is the number of chunks*/

Step 2: For all chunks (chk_1, chk_2, …, chk_n)

Step 2.1: *systemprompt* ← *SYS_PROMPT*

Step 2.2: *userprompt* ← chunk of contexts

Step 2.3: Generate the *response* by inputting the *systemprompt* and *userprompt* to the LLM to extract the concept pairs and semantic relationships between them from the input chunk of context.

Step 2.3.1: While not at the end of the generative responses

response_text ← Get the response from LLM

response ← *response* + *response_text*

Step 3: Convert the *response* into a list of concepts, named *conceptList*, where each item in the list is a dictionary with three keys: *node_1*, *node_2*, and *edge*. These keys represent two concepts and their relationship, respectively.

Step 4: *df1_KG* ← Convert the list *conceptList* to a data frame.

Step 5: For the concepts occurring in the same text chunk, add a contextual proximity relationship and save this data frame to *df2_KG*.

Step 6:

Step 6.1: Concatenate the two data frames *df1_KG and df2_KG*

Step 6.2: Group the node pairs, sum their weights, concatenate the relationships, and save the results to the *df_KG* data frame.

Step 6.3: Concatenate the unique *node_1* and *node_2* of *df1_KG and df2_KG* and store them in *nodes*.

Step 6.4: Save names and size information of *nodes* to *df_nodes* data frame.

Step7: End

Chunk of contexts

So, before we start a little bit of benefits. I think that you will hopefully see even more and understand more of the things that are beneficial when you use the ***fuzziness*** down at the end of the presentation. But, I would like to mention that they come from the aspect of looking and kind of able to express some of the things in the way how we do it. So, it means that we look at ***human intuition***, and ***human judgments*** in many cases, and many of the ***statements*** which we make are not very precise. We say that for example, with something like something large, or something is expensive. These are all imprecise ***statements*** or imprecise terms, but this is something where the ***fuzziness*** will help us to translate these fuzzy imprecise ***linguistic terms*** or phrases which we humans use into ***mathematics*** and into eventually numbers which you can apply and use in your computers or also in the ***Quantum Computing***. So, we will talk about the benefits from the aspect that we can eventually represent these ***linguistic terms*** into numbers and back. So, it also talks about imprecision so anything that is not precise not always as everything is, for example, in the case *of* ***robots***, you can say that the robot is close to some object or it's far from the object. All of the terms, close, and far would be nicely represented as fuzzy terms or fuzzy numbers and in this case could be used in the process. So, it's also relatively simple. It's not a complicated story.

Response in JSON Format


```
[
                              ⋮
 {
     "node_1": "fuzziness",
     "node_2": "mathematics",
     "edge": "helps to translate fuzzy imprecise linguistic terms into numbers",
   },
   {
     "node_1": "fuzziness",
     "node_2": "Quantum Computing",
     "edge": "helps to translate fuzzy imprecise linguistic terms into numbers that can be applied in computers or Quantum Computing",
   },
   {
     "node_1": "fuzziness",
     "node_2": "robots",
     "edge": "helps to represent imprecise terms such as close or far as fuzzy numbers, which can be used in robotics",
   },
   {
     "node_1": "fuzziness",
     "node_2": "human intuition",
     "edge": "uses human intuition to express imprecise statements or terms",
   },
   {
     "node_1": "fuzziness",
     "node_2": "human judgments",
     "edge": "relies on human judgments to express imprecise statements or terms",
   },
 {
     "node_1": "fuzziness",
     "node_2": "statements",
     "edge": "helps to translate imprecise statements into numbers",
   },
   {
     "node_1": "fuzziness",
     "node_2": "linguistic terms",
     "edge": "helps to translate fuzzy imprecise linguistic terms into mathematics and numbers",
   },
   {
     "node_1": "imprecision",
     "node_2": "robots",
     "edge": "represents imprecise terms such as close or far as fuzzy numbers, which can be used in robotics",
   },
   {
     "node_1": "imprecision",
     "node_2": "mathematics",
     "edge": "helps to translate fuzzy imprecise linguistic terms into numbers",
   },
   {
     "node_1": "imprecision",
     "node_2": "Quantum Computing",
     "edge": "helps to translate fuzzy imprecise linguistic terms into numbers that can be applied in computers or Quantum Computing",
   },
   {
     "node_1": "imprecision",
     "node_2": "statements",
     "edge": "helps to translate imprecise statements into numbers",
   },
   {
     "node_1": "imprecision",
     "node_2": "linguistic terms",
     "edge": "helps to translate fuzzy imprecise linguistic terms into mathematics and numbers",
   },
                              ⋮
]
```


Fig. 5. GAIKG example of the second chunk of contexts and the extracted *concepts* and *relationships*.

Chunk of contexts

I would like to emphasize that that's a ***fuzzy inference***, this ***process*** which you see here. So, with that, I hope that I will generate some interest in ***fuzziness*** and hopefully, you will be able to get into the details of some of the things regarding its mathematics. It is not very complicated but that allows you to and I like to kind of finish with this ***fuzziness*** you just learn a few things about it but it allows you eventually to build a ***linguistic or intelligent system*** using ***linguistic labels***. And with that, I finish, and thank you for letting me go through the whole ***process***. I hope that that was ***useful*** to understand what the ***fuzziness*** is about in a very short more less half an hour presentation. Thank you very much.

Response in JSON Format


```
[
                              ⋮
   {
     "node_1": "fuzziness",
     "node_2": "mathematics",
     "edge": "helps to translate fuzzy imprecise linguistic terms into numbers",
   },
   {
     "node_1": "fuzziness",
     "node_2": "Quantum Computing",
     "edge": "helps to translate fuzzy imprecise linguistic terms into numbers that can be applied in computers or Quantum Computing",
   },
   {
     "node_1": "fuzziness",
     "node_2": "robots",
     "edge": "helps to represent imprecise terms such as close or far as fuzzy numbers, which can be used in robotics",
   },
   {
     "node_1": "fuzziness",
     "node_2": "human intuition",
     "edge": "uses human intuition to express imprecise statements or terms",
   },
   {
     "node_1": "fuzziness",
     "node_2": "human judgments",
     "edge": "relies on human judgments to express imprecise statements or terms",
   },
   {
     "node_1": "fuzziness",
     "node_2": "statements",
     "edge": "helps to translate imprecise statements into numbers",
   },
   {
     "node_1": "fuzziness",
     "node_2": "linguistic terms",
     "edge": "helps to translate fuzzy imprecise linguistic terms into mathematics and numbers",
   },
   {
     "node_1": "imprecision",
     "node_2": "robots",
     "edge": "represents imprecise terms such as close or far as fuzzy numbers, which can be used in robotics",
   },
   {
     "node_1": "imprecision",
     "node_2": "mathematics",
     "edge": "helps to translate fuzzy imprecise linguistic terms into numbers",
   },
   {
     "node_1": "imprecision",
     "node_2": "Quantum Computing",
     "edge": "helps to translate fuzzy imprecise linguistic terms into numbers that can be applied in computers or Quantum Computing",
   },
   {
     "node_1": "imprecision",
     "node_2": "statements",
     "edge": "helps to translate imprecise statements into numbers",
   },
   {
     "node_1": "imprecision",
     "node_2": "linguistic terms",
     "edge": "helps to translate fuzzy imprecise linguistic terms into mathematics and numbers",
   },
                              ⋮
]
```


Fig. 6. GAIKG example of the last chunk of contexts and the extracted *concepts* and *relationships*.

3.3 Community Detection Mechanism of GAIKG

This subsection introduces the *community detection mechanism* of the GAIKG system. Table 3 presents the algorithm used to build a network graph that visualizes the semantics of input texts from the human and machine interaction repository.

Table 3. *Community detection mechanism* algorithm of GAIKG based on the *Ollama* platform and TAIDE model.

Input:
1. *df_KG*: A data frame containing the necessary information to create a knowledge graph, including the extracted concepts, relationships, and the weights of adjacent concepts.
2. *df_nodes*: A data frame listing the names and sizes of the extracted unique concepts.
3. *graph_KG*: a *NetwokX* knowledge graph object.
Output:
1. *knowledge graph:* A visual representation of the knowledge graph that captures the semantics of input texts from the human and machine interaction repository.
2. *df_node:* A data frame presenting the extracted concepts along with information on the name, size, and degree of each concept.
3. *no_Communities:* The number of communities detected within the graph.
4. *list_Communities:* A list containing the names of concepts within each community.
Method:
Step 1: Initialize the NetwokX knowledge graph object *graph_KG*
graph_KG ← ∅
Step 2: For all concepts and relationships in *df_KG*
Step 2.1: *Add all concepts to graph_KG*
Step 2.2: *Add all relationships to graph_KG*
Step 3: Import the networkX package and aliased it as *nx*
Step 4: Compute and measure community structure
Step 4.1: Apply the Girvan–Newman community detection algorithm using the *girvan_newman* function, and store the result in *communities_generator*.
communities_generator ← *nx.community.girvan_newman(graph_KG)*
Step 4.2: Retrieve the top level of community division from *communities_generator*.
top_level_communities ← *next* (*communities_generator*)
Step 3.4.3: Advance *communities_generator* to obtain the next level of community partitioning.
next_level_communities ← *next* (*communities_generator*)
Step 3.4.4: Sort the concepts in each community and then sort the list of communities for consistency and readability.
list_Communities ← *sorted* (map(*sorted*, *next_level_communities*))
no_Communities ← *len* (*list_Communities*)
Step 5: Assign colors to each community and store this in *df_colors*
Step 6: Map and assign the degree of each concept in *graph_KG* to the corresponding concept in *df_node*
Step 7: Use *graph_KG* to create a visual knowledge graph that represents the semantics of the input texts.
Step 8: End

Fig. 7 displays the *knowledge graph* created from the extracted *concepts* and *relationships* presented in Figs. 5 and 6. The nodes represent individual *concepts*, while the edges indicate the *relationships* between them. The graph encompasses concepts such as *fuzziness*, *fuzzy inference*, *human judgments*, *quantum computing*, *linguistic terms*, *mathematics*, and several others. The node size denotes the concept's importance, with *fuzziness* being the largest and most central or significant *concept*. The edge width varies, indicating the strength of the *relationships*; for example, *thicker lines suggest stronger relationships*. Notably, *fuzziness* is strongly connected to *mathematics*, *imprecision*, and *quantum computing*.

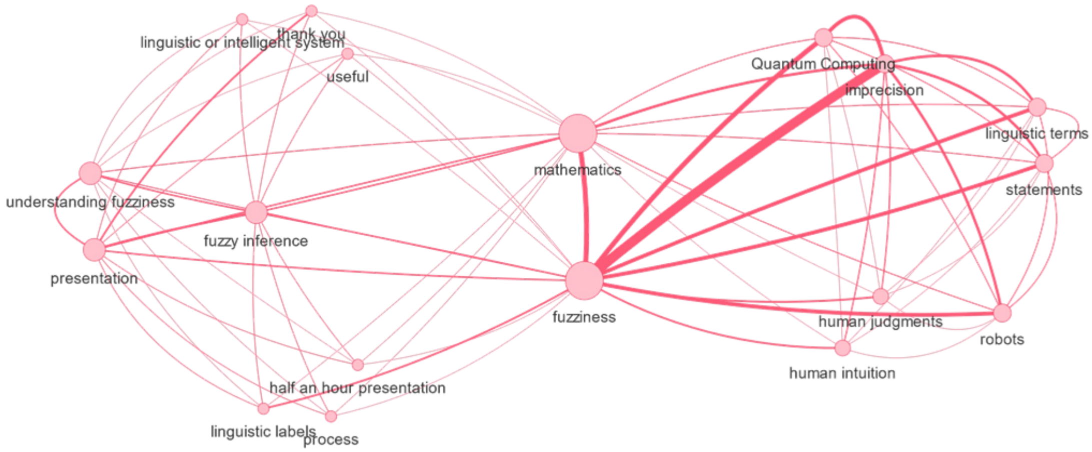


Fig. 7. *Knowledge graph* for the second and last chunks of GAIKG based on the TAIDE model.

Table 4 provides detailed information on the *knowledge graph*, including the names of *concepts* (columns *node_1* and *node_2*), texts of *relationships* (column *edge*), and the weighting of two concepts (column *weight*) derived from the texts of the second and the final chunk. Eighteen communities have been extracted from the captions of video 1. The concepts of this community include *fuzziness*, *Quantum Computing*, *human intuition*, *human judgments*, …, and *useful*.

Table 4. Information on the knowledge graph of GAIKG based on the TAIDE model.

node_1	node_2	No.	edge	weight
fuzziness	*Quantum Computing*	2	helps to translate fuzzy imprecise linguistic terms into numbers that can be applied in computers or Quantum Computing, contextual proximity	**4.5**
fuzziness	*human intuition*	2	uses human intuition to express imprecise statements or terms, contextual proximity	**2.75**
fuzziness	*human judgments*	2	relies on human judgments to express imprecise statements or terms, contextual proximity	**2.75**
fuzziness	*imprecision*	2	contextual proximity	**8.75**
fuzziness	*linguistic terms*	2	helps to translate fuzzy imprecise linguistic terms into mathematics and numbers, contextual proximity	**4.5**
fuzziness	*robots*	2	helps to represent imprecise terms such as close or far as fuzzy numbers, which can be used in robotics, contextual proximity	**4.5**
fuzziness	*statements*	2	helps to translate imprecise statements into numbers, contextual proximity	**4.5**
fuzziness	*mathematics*	2	helps to translate fuzzy imprecise linguistic terms into numbers, related to, contextual proximity	**6.5**
fuzziness	*linguistic labels*	19	can be used to build, contextual proximity	**1.75**
fuzziness	*fuzzy inference*	19	contextual proximity	**1.5**
fuzziness	*presentation*	19	contextual proximity	**1.5**
fuzziness	*understanding fuzziness*	19	contextual proximity	**1**
fuzziness	*half an hour presentation*	19	contextual proximity	**0.5**
fuzziness	*linguistic or intelligent system*	19	contextual proximity	**0.5**
fuzziness	*process*	19	contextual proximity	**0.5**
fuzziness	*thank you*	19	contextual proximity	**0.5**
fuzziness	*useful*	19	contextual proximity	**0.5**

4. Taiwanese/English Co-Learning based on Quantum CI Model with GAI Image

This section describes the Taiwanese/English co-learning based on the Quantum CI (QCI) model with GAI Image. Section 4.1 introduces a *multi-modal GAI Interactive Learning* (GAIIL) *platform*. The GAIIL-based QCI model optimization with validation mechanism and the *Quantum Fuzzy Inference Mechanism* (QFIM) of the QCI model are presented in Sections 4.2 and 4.3, respectively.

4.1 Multimodal GAI Interactive Learning (GAIIL) Platform

Fig. 8 shows a multimodal *GAI Interactive Learning* (GAIIL) platform that exemplifies the integration of multimodal data types. It is designed to facilitate human-machine interaction by using GAI technologies to generate images, texts, or voice outputs from various inputs, enabling co-learning of the Taiwanese/English language between learners and machines. During the input phase, learners can choose from three options: 1) ***text type***: allowing for English or Chinese text inputs; 2) ***voice type***: enabling English, Chinese, or Taiwanese voice inputs; and 3) ***image type***.

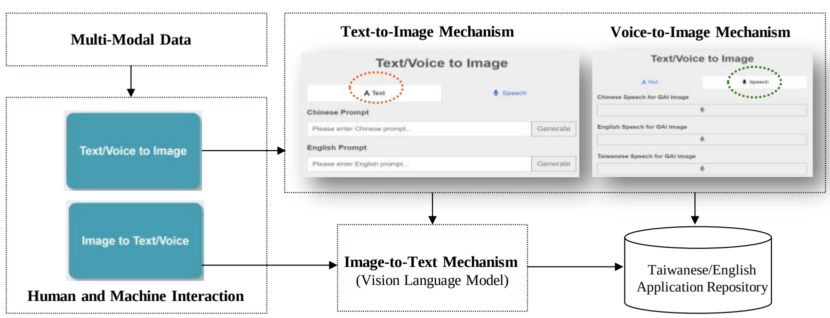


Fig. 8. *Multimodal GAI Interactive Learning (GAIIL) platform for Taiwanese/English Co-Learning.*

Upon receiving text data for the prompts, the adopted *image generation AI model* (Espinoza-Guerra, 2024) within the *Text-to-Image* and *Voice-to-Image* mechanisms creates visual representations and generates the corresponding texts and voices using the *text translation model* (Liam, 2024) and *speech recognition technology*. Moreover, the *Meta AI UST model* is used to transform between English and Taiwanese voices (Chen et al., 2022; Lee et al., 2024b). In addition to its *text-to-image* capabilities, the GAIIL platform can also generate textual or spoken outputs in different languages through *content summarization* from the input image, utilizing the *generative vision-language model* (Kim et al., 2024) and the *text translation model*. This offers learners an immersive experience by allowing them to *visualize descriptions* or *summarize visual scenarios*. Finally, the data of Taiwanese/English applications are stored in the repository for optimizing and validating the GAIIL-based QCI model. For detailed information on the user interface, please refer to Item No. 1 in the Appendix.

4.2 GAIIL-based QCI Model Optimization and Validation Mechanism

This section introduces the evaluation mechanism based on *Human Intelligence* (HI), as well as GAIIL-based QCI model optimization and validation mechanisms, all based on the QCI model with GAI Image for Taiwanese/English co-learning. We design the *Taiwanese/English co-learning* with GAIIL-based QCI model for elementary school students as follows: In Taiwan, the elementary school consists of 6 grades, categorized into *low-grade* (1st and 2nd), *middle-grade* (3rd and 4th), and *high-grade* (5th and 6th) levels. At the *low-grade* level, students experience the GAIIL-based QCI model through interaction with the *QCI&GAI robot* or by using other *learning tools*. *Middle-grade* students are introduced to Blockly logic design and interact with the *QCI&GAI robot* for interactive Taiwanese/English language learning and data collection. *High-grade* students utilize the collected data to build, train, validate, and optimize the GAIIL-based QCI model.

The PSO (Lee et al., 2018) method mimics the behavior of social insects and particles to solve complex problems. The GANN (Lee et al., 2023c) method trains neural networks using genetic algorithms. Both PSO and GANN are types of evolutionary algorithms, and EC is one of the important pillars of CI. To help young students learn the basic concepts of EC during the QCI sandbox events in 2023 and 2024, we chose PSO and GANN methods to allow them to learn these concepts and refine their models using their own collected data. The CI model is refined through PSO and GANN methods by inputting the training data and target data to perform regression and generate the trained model when one of the terminal conditions is achieved. After that, we load the best-optimized model to infer the results using classical fuzzy inference and quantum fuzzy inference and then compare their performance with the before-learning CI model.

The GAIIL platform assists in Taiwanese/English language learning and teaching, as well as evaluates student learning outcomes based on the GAIIL-based QCI model to achieve the goal of GAI-assisted language learning and teaching. This paper collects two types of datasets, **DS1** and **DS2**, with the following descriptions:

- **DS1:** The involved subjects engage in voice-to-image activities by speaking English or Chinese to generate corresponding images via the multimodal GAIIL platform. The recognized texts from the received speech are used to generate the corresponding images. Then, humans evaluate four qualities of the GAIIL platform, including (1) *recognized English texts* (*ENTxt*), (2) *recognized Chinese texts* (*CNTxt*), (3) *images by speaking English* (*ENImg*) and (4) *images by speaking Chinese* (*CNImg*), as well as label them with different levels from A to D, where *level A* is the best. Based on these criteria, we convert the human linguistic language to crisp values by randomizing it as follows: *Level A*: (0.75, 1], *Level B*: (0.5, 0.75], *Level C*: (0.25, 0.5], and *Level D*: [0, 0.25].
- **DS2:** The subjects also speak Taiwanese to generate the corresponding image. Then, humans evaluate the quality of the *generated English texts* (*ENTxt*), *translated Chinese texts* (*CNTxt*), and *generated images* (*TWImg*) on four levels, too.

For **DS1**, the input *fuzzy variables* of the GAIIL-based QCI model are *ENTxt*, *ENImg*, *CNTxt*, and *CNImg*, each with three *fuzzy linguistic terms* (*Bad*, *Normal*, and *Good*). The output *fuzzy variable* is *GAIModel* with five *fuzzy linguistic terms* (*VeryBad*, *Bad*, *Normal*, *Good*, *VeryGood*), representing the quality of the GAIIL-based QCI model. The total number of *fuzzy inference rules* is 81. Using **DS1**, we compare the differences between *CI inference* and *QCI inference*, train the model based on PSO and GANN using a portion of **DS1**, and validate the GAIIL-based QCI model using the remaining **DS1**. For **DS2**, we analyze the performance of *ENTxt*, *CNTxt*, and *TWImg* in statistics.

4.3 Quantum Fuzzy Inference Mechanism of QCI Model

This section describes the *Quantum Fuzzy Inference Mechanism* (QFIM) of the QCI model based on the *Quantum Fuzzy Inference Engine* (QFIE) proposed by Acampora et al. (2023). Fig. 9 shows the flowchart detailing the QFIM associated with the QCI model. It showcases a synergistic blend of *fuzzy logic* and *quantum computing* that generates a QCI model from an input CI model. The QFIM is integrated into the developed *QCI&AI-FML learning platform* to generate *quantum circuits* for the QCI model, which is helpful for teaching and learning QCI to young students.

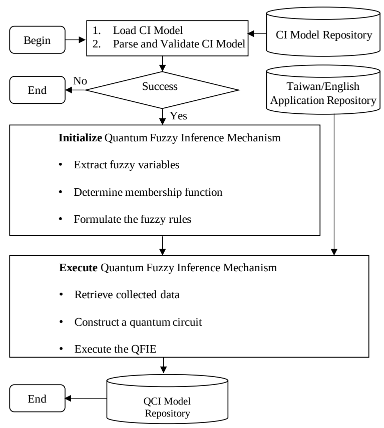


Fig. 9. Quantum fuzzy inference mechanism with the QCI model flowchart for *quantum circuit* generation.

In *Application 2* of this paper, with four *input fuzzy variables* and three *fuzzy terms* each, a total of 81 *fuzzy rules* are required to cover every possible combination of inputs. First, the QFIM loads the constructed CI model provided by the learners through the *QCI&AI-FML learning platform*. Then, it parses and validates the CI model. If it successfully passes validation, the QFIM begins the initialization and execution processes for *quantum fuzzy inference*. This includes formulating and mapping *fuzzy rules* onto a *quantum circuit*, and finally, executing the *inference process* of QFIM to generate a *quantum circuit*, the estimated count, and probability after executing the *quantum circuit* for a certain number of shots. Table 5 provides a structured view of how binary strings correspond to *fuzzy linguistic terms* for various *fuzzy variables* in **DS1** of *Application 2*. The *fuzzy linguistic terms*, such as *Bad*, *Normal*, and *Good*, are assigned specific binary representations that serve as the bridge between the *fuzzy logic* and the QCI model.

Table 5. Mapping of binary strings to fuzzy linguistic terms for each fuzzy variable in **DS1** of Application 2.

ENTxt	*ENImg*	*CNTxt*	*CNImg*	*GAIModel*
Bad → 00	*Bad* → 00	*Bad* → 00	*Bad* → 00	*VeryBad* → 00001
Normal → 01	*Normal* → 01	*Normal* → 01	*Normal* → 01	*Bad* → 00010
Good → 10	*Good* → 10	*Good* → 10	*Good* → 10	*Normal* → 00100
				Good → 01000
				VeryGood → 10000

Fig. 10 illustrates the mapping from *fuzzy rules* to *oracle-based fuzzy rules* using Eq. (1). The function *f* symbolizes a rule converting a set of binary inputs into a binary output. Each *fuzzy rule* relates the states of input *fuzzy variables*, including *ENTxt*, *ENImg*, *CNTxt*, and *CNImg*, to an output state for the *fuzzy variable GAIModel*. Each *fuzzy rule* defines an outcome for *GAIModel* (ranging from *VeryBad* to *VeryGood*) based on the combination of the states of the input *fuzzy variables*.

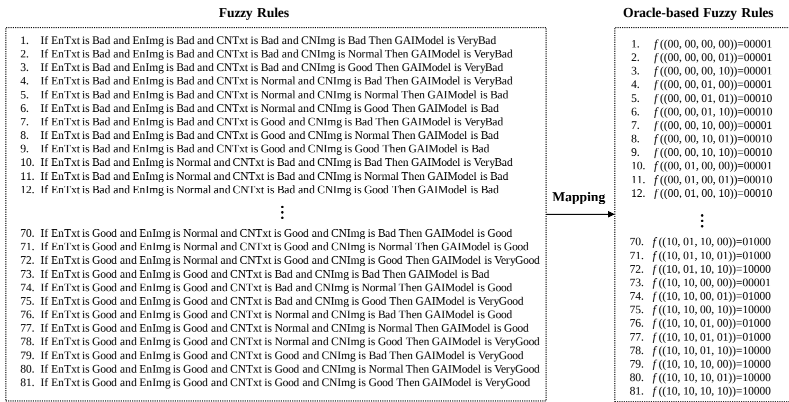


Fig. 10. Oracle view of the partial fuzzy rules for QFIM of QCI model.

$$f\colon \mathcal{A}_{\mathcal{S}} \rightarrow \mathcal{C}_{\mathcal{S}} \qquad (1)$$

where (1) *f* is an oracle that maps the antecedent parts to the consequent parts to compose fuzzy rules, (2) $\mathcal{A}_{\mathcal{S}}$ is a Cartesian product that contains the binary encoding of all possible antecedents that can be defined with the input variables, and (3) $\mathcal{C}_{\mathcal{S}}$ contains the binary encoding of all possible consequent parts that can be defined with the output variable (Acampora et al., 2023).

Fig. 11 depicts the implementation of *oracle-based fuzzy rules* from rule 1 (R1) to rule 12 (R12) as a *quantum circuit* where 1) the *X gates* represent *quantum NOT operations* that flip the state of a qubit, 2) $\oplus$ stands for the *C-NOT operation quantum gate*, and 3) *X quantum gates* are applied symmetrically with respect to the *Multi-Controlled X (MCX) gate*. In Fig. 11, the *quantum registers* are comprised of individual qubits, which are labeled as *ENTxt*, *ENImg*, *CNTxt*, *CNImg*, and *GAIModel*. Each label represents a distinct aspect of the *fuzzy variables* that are being processed. Consider the set {*ENTxt*: 0.7, *ENImg*: 0.64, *CNTxt*: 0.47, *CNImg*: 0.97} as an example. These values are used as input for the QFIM of the QCI model.

- After executing the constructed *quantum circuit* on the *IBM Qasm Simulator*, the membership degrees stored in the state of the *quantum registers* are {*ENTxt*: [0.0, 0.50, 0.50], *ENImg*: [0.0, 0.79, 0.20], *CNTxt*: [0.0, 1.0, 0.0], *CNImg*: [0.0, 0.0, 1.0]}. The encoded amplitudes of the states are |ψ⟩ = [0, 0.707, 0.707, 0], |ψ⟩ = [0, 0.894, 0.447, 0], |ψ⟩ = [0, 1, 0, 0], and |ψ⟩ = [0, 0, 1, 0] for *ENTxt*, *ENImg*, *CNTxt*, and *CNImg*, respectively.

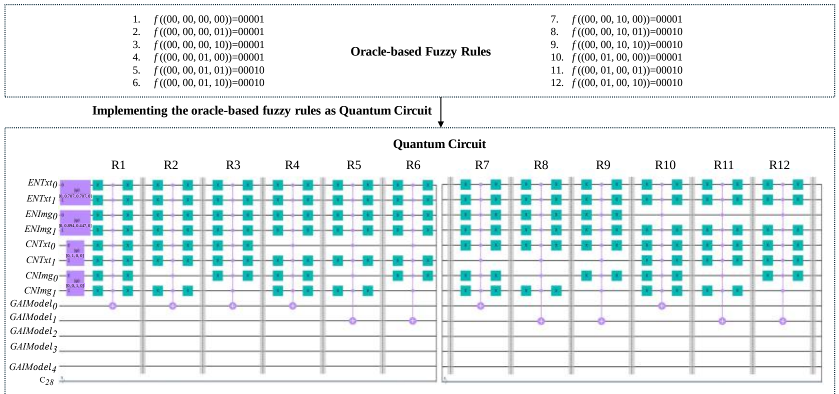


Fig. 11. Implementing the oracle-based *fuzzy rules* as a *quantum circuit*.

- The output probability for the *fuzzy sets* of the *GAIModel*, which include *VeryBad* (00001), *Bad* (00010), *Normal* (00100), *Good* (01000), and *VeryGood* (10000), is given as {'00001': 0, '00010': 0, '00100': 0.40175, '01000': 0.498, '10000': 0.10675}. These cut *membership functions* were then aggregated to obtain the output *fuzzy set GAIModel* = 0.591, where the *defuzzification operator* was applied. Finally, if *ENTxt* (0.7) is *Good*, *ENImg* (0.64) is *Normal*, *CNTxt* (0.47) is *Normal*, and *CNImg* (0.97) is *Good*, then *GAIModel* (0.591) is determined to be *Good*.

5. Experimental Results

This section introduces the performance of integrating QCI with GAI for co-learning Taiwanese and English, based on the construction of a TAIDE-based *knowledge graph* and multimodal data transformation. Sections 5.1 to 5.3 present the experimental results for *Application 1*. Section 5.1 introduces the sources of the experimental data. In Sections 5.2 and 5.3, we use the collected QCI data to compare the performance of the GAIKG based on the number of *concepts* and *relations* between the TAIDE model and the *Ollama* platform. Sections 5.4 and 5.5 present the experimental results of *Application 2*. The optimized models based on PSO and GANN methods are compared in Section 5.4. Section 5.5 presents the difference between the QCI model and the CI model before and after optimization.

5.1 *Application 1: Sources of the Experimental Data*

Fig. 12 shows the sources of the experimental data for *Application 1*, which were collected at the *QCI Sandbox for Pre-University and Undergraduate Students*, organized by the IEEE CIS Education Portal Subcommittee / IEEE R10 EAC and SPNIC in 2023 and 2024.

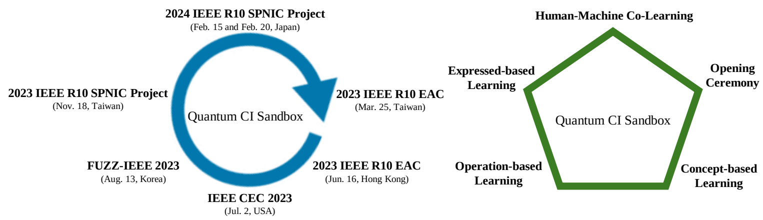


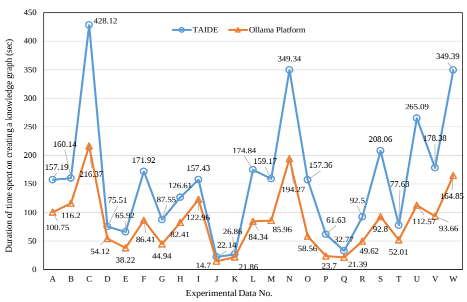
Fig. 12. Experimental text data sets of six IEEE education events in 2023 and 2024.

The total number of experimental *text data sets* is 23, which comes from six *IEEE education events*. Each event is composed of five sessions, including the *Opening Ceremony*, *Concept-based Learning*, *Operation-based Learning*, *Expressed-based Learning*, and *Human-Machine Co-learning*. We used the captions of the videos recorded during the event or edited for the event to extract the *concepts* and *relations* between two concepts to create the *knowledge graph* for the *QCI Sandbox for Pre-University and Undergraduate Students*.

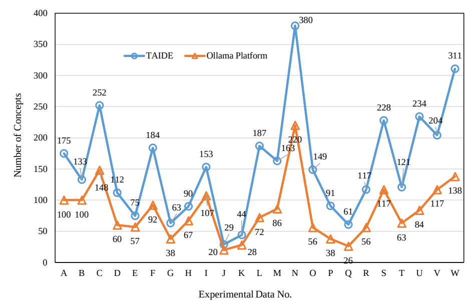
5.2 *Application 1: Performance Comparison of GAIKG between the TAIDE Model and Ollama Platform*

This section presents a comparison of the performance of GAIKG between the TAIDE model and the *Ollama* platform based on the 23 *text Data Sets* from the six *IEEE education events* in 2023 and 2024. The details of the *Data Set* A to *Data Set* W are listed in Item No. 4 in the Appendix. The methodology for extracting concepts and relations is detailed in Section 3.2. It involves executing Python code for each dataset within a GPU environment (Geforce 4080). During execution, information such as the time spent creating a *knowledge graph* and the number of concepts, relations, and communities included in the resulting *knowledge graph* is stored. This approach allows us to compare the LLM TAIDE and *Ollama* platform based on these four aspects.

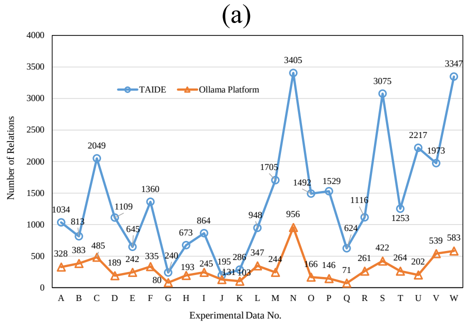
Fig. 13(a)-(d) shows the duration of time spent creating a *knowledge graph*, and the number of *concepts*, *relations*, and *communities* included in the created *knowledge graph*, respectively, between the TAIDE model and the *Ollama* platform. The *Ollama* platform is more time-efficient in creating *knowledge graphs*, while the TAIDE model generates more *concepts* and *relationships*, suggesting a more comprehensive and nuanced *knowledge graph*. Conversely, the *Ollama* platform is better at identifying distinct *communities*, implying superior categorization capabilities. Thus, the choice between TAIDE and *Ollama* may depend on the specific needs for efficiency, detail, or clarity in *knowledge graph* creation.



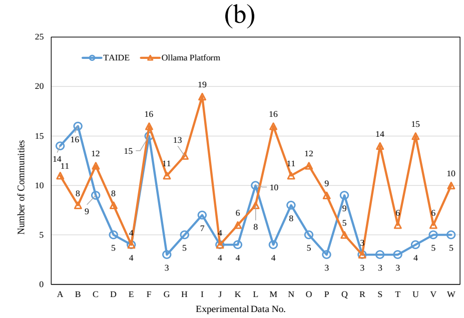
(a)



(b)



(c)



(d)

Fig. 13. (a) Duration of time spent and the number of (b) *concepts*, (c) *relations*, and (d) *communities*.

5.3 Application 1: Analysis of Concept-based Learning on Fuzzy Systems

This section analyzes the differences between the TAIDE model and the *Ollama* platform in the generated *knowledge graph* using experimental *Data Set* B extracted from the captions of *video 1* by Prof. Marek Reformat at the CcSR 2024 workshop in Japan. Figs. 14 (a)-(b) show the *knowledge graphs* (KGs) generated by the TAIDE model and *Ollama* platform. It indicates that the *concept* with the greatest centrality in the KG by the TAIDE model is *temperature*, measured at *34 degrees*, whereas in the KG by the *Ollama* platform, the most central concept is *price*, measured at *25 degrees*. The KG generated by the TAIDE model is characterized by dense interconnections among its *concepts*, suggesting a complex and integrated structure of closely associated *concepts*. In contrast, the KG from the *Ollama* platform exhibits distinct *community* boundaries, reflecting a clearer and more segmented conceptual organization.

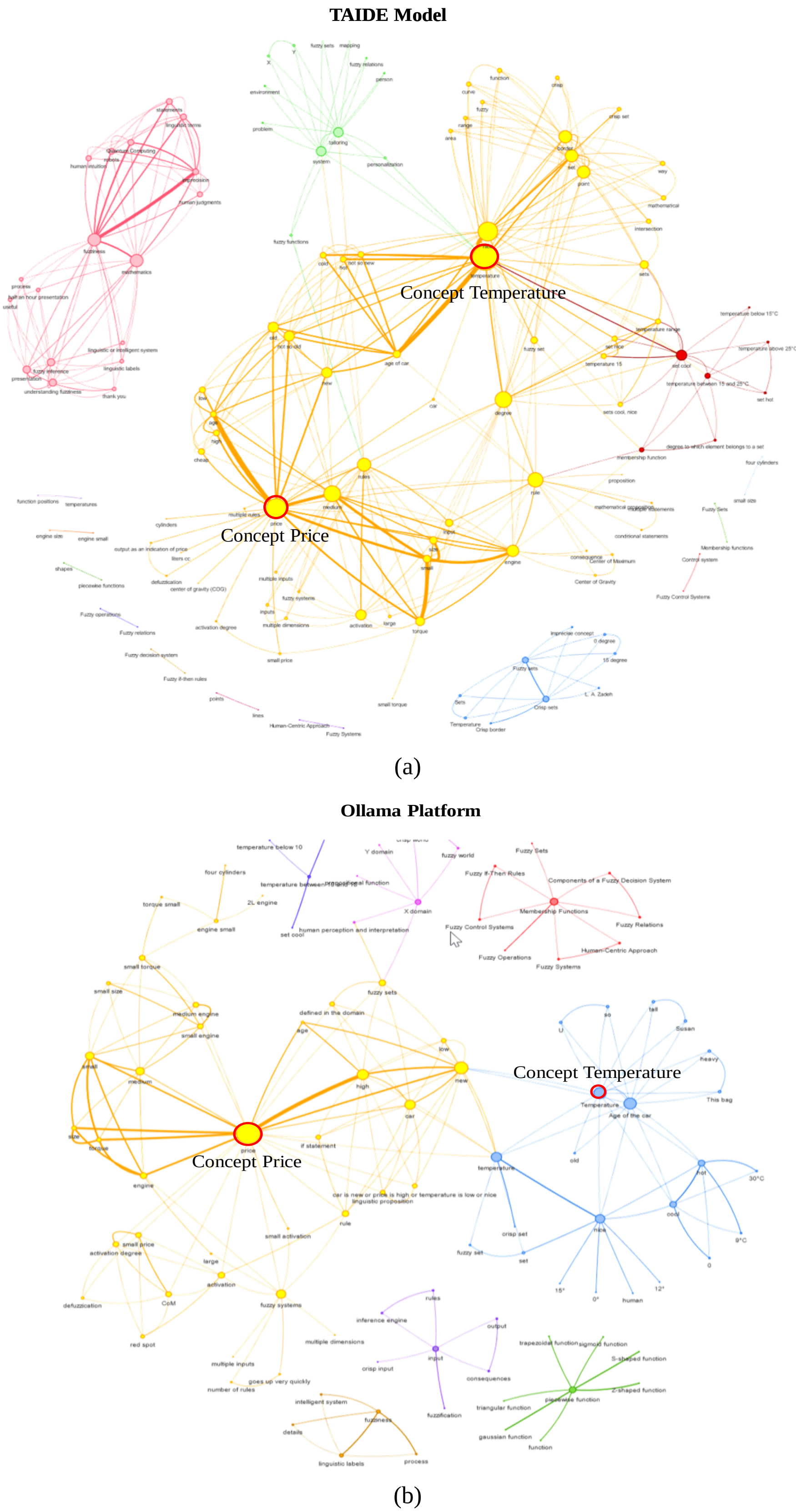


Fig. 14. *Knowledge graph* generated from *Data Set* B by the (a) TAIDE and (b) *Ollama* via the GAIKG.

5.4 Application 2: Optimized models based on PSO and GANN mechanisms

In February 2024, nine learners from different countries participated in the experiment for *Application 2* using the GAIIL platform for Taiwanese/ English co-learning: six from Taiwan, two from China, and one from Malaysia. Subsequently, we collected Data Sets 1 and 2 (**DS1** and **DS2**), the details of which are provided in Section 4.2. The spoken texts in English and Chinese versions can be referred to in the published paper (Lee et al., 2024a). The total number of DS1 is 144, with 96 as training data collected in Taiwan and 48 as testing data collected in Japan. For the DS2, the total number of data is 96. We adopted two learning mechanisms, PSO and GANN, to optimize the QCI model.

Table 6 outlines the experimental steps. Table 7 shows the MSE, accuracy, and training time values for different combinations of particle number and iteration number optimized by the PSO learning mechanism. The values with a star and in bold denote the lowest MSE considering the same particle numbers but different iterations, and the value with a gray background represents the best-optimized QCI model. Fig. 15 displays the MSE curve and accuracy bar for the lowest MSE obtained from Table 7, with the optimized QCI model being the combination of 50 particles and 2000 iterations.

Table 6. Experimental steps using (a) PSO learning mechanism and (b) GANN learning mechanism.

(a)	
Step 1.	Set the number of particles to 10.
Step 2.	Optimize the model using 10 particles and 100 iterations.
Step 3.	Evaluate the trained model using MSE and accuracy, which measures semantic matching results.
Step 4.	Store the best model based on the lowest MSE.
Step 5.	Repeat Steps 1 to 4 with the number of particles set to 20, 30, 40, and 50.
Step 6.	Select the model with the lowest MSE from the results of Steps 4 and 5.
Step 7.	Use the model with the lowest MSE from Step 6 as the optimized model.
(b)	
Step 1.	Configure the GANN architecture with four input nodes in the input layer, 12 nodes in the first hidden layer, 4 nodes in the second hidden layer, and one node in the output layer.
Step 2.	Set the selection method as rank selection, the crossover method as a two-point crossover, and the mutation method as random mutation.
Step 3.	Set four different combinations of crossover rate and mutation rate (p_c, p_m): (0.9, 0.1), (0.9, 0.05), (0.6, 0.1), and (0.6, 0.05).
Step 4.	Optimize the model using 100 iterations for each of the four combinations.
Step 5.	Evaluate the trained model using MSE.
Step 6.	Store the best model based on the lowest MSE.
Step 7.	Repeat Steps 1 to 6 with the number of iterations set to 200, 500, 1000, and 2000.
Step 8.	Select the model with the lowest MSE from the results of Steps 6 and 7.
Step 9.	Use the model with the lowest MSE from Step 8 as the optimized model.

Table 7. MSE and accuracy values for different combinations of particle number and iteration number.

Iterations / Particles	MSE / Accuracy / Training Time (sec)				
	100	200	500	1000	2000
10	0.00516/0.91/7	0.00452/0.9/14	0.00511/0.85/37	**0.00417***/0.89/74	0.00425/0.83/145
20	0.00505/0.86/14	0.00475/0.8730	0.00442/0.8979	0.00392/0.87/152	**0.00376***/0.87/294
30	0.00544/0.86/22	0.00399/0.9/40	**0.00363***/0.91/105	0.00388/0.93/202	0.00367/0.9/412
40	0.00445/0.9/30	0.00417/0.92/64	0.004/0.9/164	0.00392/0.88/312	**0.00364***/0.83/603
50	0.00474/0.87/42	0.00427/0.91/81	0.00406/0.9/197	0.00369/0.86/84	**0.00349***/0.91/753

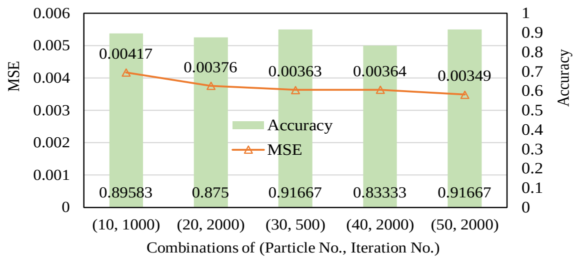


Fig. 15. MSE curve and accuracy bar for the lowest MSE obtained under different iteration numbers using PSO.

Fig. 16. MSE values for before-learning model and optimized models by PSO and GANN.

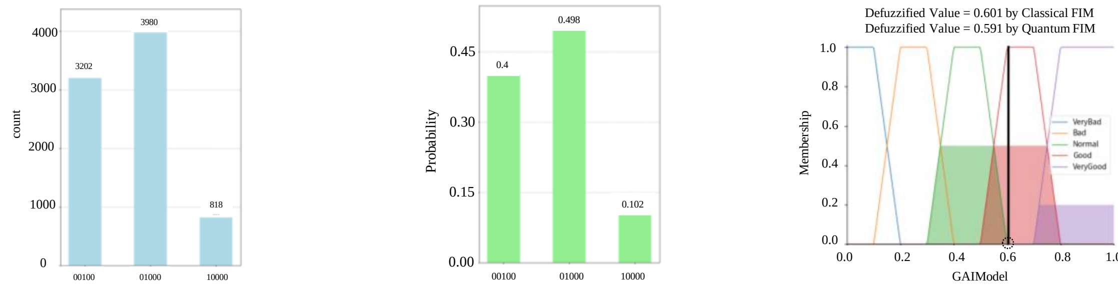
Table 8 shows MSE and training time for different combinations of p_c, p_m, and iteration number optimized by the GANN learning mechanism. The values with a star and in bold denote the lowest MSE considering the same iteration numbers but different combinations of crossover and mutation rates, and the value with a gray background represents the best-optimized QCI model with p_c=0.9, p_m=0.05, and 500 iterations. Fig. 16 shows the MSE values for the before-learning model and optimized QCI models by PSO and GANN for the training and testing data sets. It indicates that the QCI model optimized by PSO performs better than the one optimized by GANN. However, the training time of GANN (44.24 sec) is shorter than that of PSO (753 sec).

Table 8. MSE and training time for different combinations.

Iterations / (p_c, p_m)	MSE / Training Time (sec)				
	100	200	500	1000	2000
(0.9, 0.1)	0.01313/70.1	0.01324/103.0	0.01195/56.8	0.01400/90.0	0.01168/160.77
(0.9, 0.05)	0.01328/34.3	**0.01194***/60.6	**0.01033***/44.24	**0.01213***/92.38	0.01155/172.02
(0.6, 0.1)	0.01317/10.65	0.01287/19.1	0.01282/55.25	0.01278/95.0	0.01214/158.82
(0.6, 0.05)	**0.01262***/52.57	0.01251/76.7	0.01296/62.6	0.01255/95.76	**0.01058***/157.55

5.5 Application 2: Difference between the QCI Model and the CI Model before and after Optimization

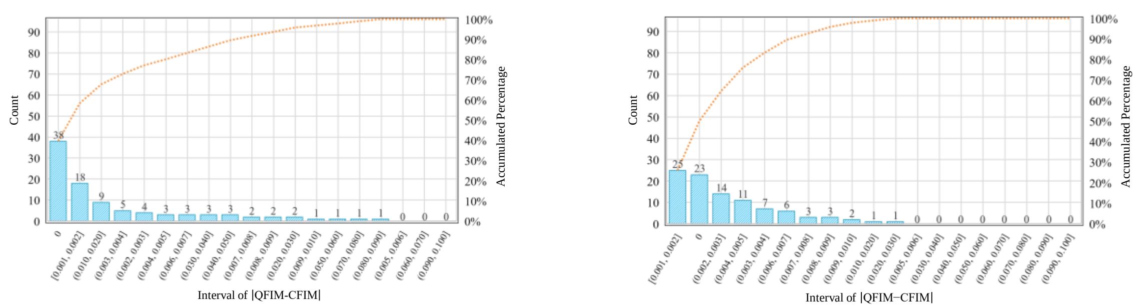
This section utilizes **DS1** as the input data to implement a quantum fuzzy inference mechanism with 8,000 shots and evaluate the average performance difference compared to classical fuzzy inference. For example, consider the values {*ENTxt*: 0.7, *ENImg*: 0.64, *CNTxt*: 0.47, *CNImg*: 0.97}. Fig. 17(a)-(c) respectively displays the count, probability, and defuzzified value for the fuzzy sets of the *GAIModel* using the 29th data point as an example.



(a) (b) (c)

Fig. 17. (a) count, (b) probability, and (c) defuzzified value

Fig. 17(c) shows there is a very small difference between the CI model and the QCI model. Fig. 18 presents the Pareto chart for the difference between QFIM and classical fuzzy inference mechanism (CFIM) using the before-learning model and the best-optimized models by PSO. It suggests that the optimization through PSO has effectively refined the model, leading to a reduction in the variability of the differences between the QFIM and CFIM outcomes. This refinement is highlighted by the more distinct peak at the smallest interval and the steeper Pareto curve.



(a) (b)

Fig. 18. Pareto chart using (a) before-learning model and (b) best-optimized models by PSO.

6. Conclusion

This paper presents a QCI model combined with GAI for learning both Taiwanese and English languages through human-machine interactions. The QCI model comprises two phases: human-machine interfaces and detailed data analysis for quantum circuit generation and real-world applications, which significantly enhance student engagement and streamline the learning process. The synergy between HI and MI creates an effective teaching environment. The use of the *Ollama* platform and the TAIDE model facilitates the creation of a knowledge graph, essential for understanding the basic concepts of CI and quantum CI, as well as for sharing learners' experiences. Although quantum technology is a fundamental building block in this paper, its integration with CI and GAI demonstrates the proposed QCI model's innovation in merging quantum computing with educational tools, making it a valuable resource for high school and university students. In the future, we will employ graph neural networks for classification and explore the similarities among knowledge graphs generated by various categories, such as educational levels (elementary school, junior high school, senior high school, and universities), countries, motivation levels (high, middle, or low) obtained from questionnaires, and usage hours of the QCI&AI learning tools. Moving forward, our goal is to refine and expand this model to reach a wider audience, thereby contributing to global educational progress in quantum and artificial intelligence learning.

Appendix

In this Appendix, the short textual descriptions of the videos are listed in Table 9.

Table 9. Short textual descriptions.

Item No.	Short Textual Descriptions
1	**Topic: Development between humans & machines and Introduction to QCI Agent & GAI Image Platform** **Link:** https://sites.google.com/asap.nutn.edu.tw/ai-fml-international-academy/activities-in-taiwan/activity-2024/Integrating_QCI_GAI/GAI
2	**Topic: Introduction to QCI&AI-FML Learning Platform** **Link:** https://sites.google.com/asap.nutn.edu.tw/ai-fml-international-academy/activities-in-taiwan/activity-2024/Integrating_QCI_GAI/QCI
3	**Topic: Introduction to Fuzzy Systems** **Link:** https://youtu.be/HMzSuGXmv_E The speaker offers an accessible introduction to fuzzy systems, explaining their application in human-machine interactions and decision-making processes. Key concepts like fuzzy sets, membership functions, and fuzzy rules are outlined, emphasizing the translation of human intuition into computational terms. By using practical examples, such as temperature categorization and car valuation, the speaker demonstrates how fuzzy logic copes with imprecision, leading to more adaptable and personalized systems. The speaker concludes by hoping to spark further interest in the potential of fuzzy logic to build intelligent systems that emulate human reasoning.
4	**Topic: Generated Knowledge Graphs from Data Set A to Data Set W** Link: https://sites.google.com/asap.nutn.edu.tw/qcimodel-gai-knowledge-graph/gaikg-with-the-taide-model-and-ollama-platform

Acknowledgments

The authors would like to thank the financial support sponsored by the National Science and Technology Council (NSTC) of Taiwan under the grants 112-2622-E-024-002 and 112-2221-E-024-007-MY2. Additionally, the authors would like to thank Pei-Yu Wu and Szu-Chi Chiu for their video editing, as well as the faculty and students involved in the events and experiments.

Reference

Acampora G, Schiattarella R, Vitiello A (2023) On the Implementation of Fuzzy Inference Engines on Quantum Computers. IEEE Transactions on Fuzzy Systems, 31 (5): 1419-1433.

Acampora G, Vitiello A (2023) An IEEE Std 1855 driver for synthetizing quantum fuzzy inference engines. 2023 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2023), Incheon, Korea, Aug. 13-17.
Acampora G (2024) Quantum computing meets artificial intelligence. https://youtu.be/gQXWddPGpEo.
Baldassarre MT, Caivano D, Nieto BF, Gigante D, Ragone A (2024) The social impact of generative AI: an analysis on ChatGPT. https://arxiv.org/abs/2403.04667.
Chen PJ, Tran K, Yang Y, Du J, Kao J, Chung YA, Tomasello P, Duquenne PA, Schwenk H, Gong H, Inaguma H, Popuri S, Wang C, Pino J, Hsu WN, Lee A (2022) Speech-to-speech translation for a real-world unwritten language. https://arxiv.org/abs/2211.06474.
IEEE CIS (2016) 1855-2016-IEEE Standard for Fuzzy Markup Language. https://ieeexplore.ieee.org/document/7479441.
Ishibuchi H (2019) AI and CI. IEEE Computational Intelligence Magazine 14(3): 2.
Lee CS, Wang MH, Yen SJ, Wei TH, Wu IC, Chou PC, Chou CH, Wang MW, Yang TH (2016) Human vs. computer Go: review and prospect. IEEE Computational Intelligence Magazine, 11(3): 67-72.
Lee CS, Wang MH, Wang CS, Teytaud O, Liu JL, Lin SW, Hung PH (2018) PSO-based fuzzy markup language for student learning performance evaluation and educational application. IEEE Transactions on Fuzzy Systems, 26(5):2618-2633.
Lee CS, Wang MH, Tsai YL, Ko LW, Tsai BY, Hung PH, Lin LA, Kubota N (2019) Intelligent agent for real-world applications on robotic edutainment and humanized co-learning. Journal of Ambient Intelligence and Humanized Computing 11: 3121-3139.
Lee CS, Wang MH, Ko LW, Lee YH, Ohashi H, Kubota N, Nojima Y, Su SF (2020) Human intelligence meets smart machine: a special event at the IEEE International Conference on Systems, Man, and Cybernetics 2018. IEEE Systems, Man, and Cybernetics Magazine, 6(1): 23-31.
Lee CS, Wang MH, Chang RP, Liu HC, Chiu SC, Chang YC, Lin LA, Chen SC (2022) Computational intelligence and AI-FML experience model for pre-university student learning and practice. The 18th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2022), Kitakyushu, Japan, Dec. 16-18, 2022.
Lee CS, Wang MH, Chen CY, Liang CC, Yang SC, Horng MF (2023a) PSO-based CI agent with learning tool for student experience in real-world application. The 15th International Conference on Genetic and Evolutionary Computing (ICGEC 2023), Kaohsiung, Taiwan, Oct. 6-8.
Lee CS, Wang MH, Reformat M, Huang SH (2023b) Human intelligence-based Metaverse for co-learning of students and smart machines. Journal of Ambient Intelligence and Humanized Computing, 14: 7695-7718.
Lee CS, Wang MH, Chen CY, Yang FJ, Dockhorn A (2023c) Genetic assessment agent for high-school student and machine co-learning model construction on computational intelligence experience. 2023 IEEE Congress on Evolutionary Computation (IEEE CEC 2023), Chicago, USA, Jul. 1-5.
Lee CS, Wang MH, Chiang JK, Wu PY, Chiu SC, Yang SC, Kubota N. Siow CZ, Sato-Shimokawara E, Nojima Y, Acampora G (2024a) Quantum computational intelligence with generative AI image for human-machine interaction. 2024 IEEE Congress on Computational Intelligence (IEEE WCCI 2024), Yokohama, Japan, June 30-July 5.
Lee CS, Wang MH, Chen CY, Lee YH, Tseng GY, Yang SC, Reformat M (2024b) Transformer-based semantic SBERT robot with CI mechanism for students and machine co-learning. 2024 IEEE Congress on Computational Intelligence (IEEE WCCI 2024), Yokohama, Japan, June 30-July 5.
Liam J (2024) Trans-opus-mt-zh/en-en/zh. https://huggingface.co/liam168.
OASE Lab. (2019) AI-FML International Academy. https://sites.google.com/asap.nutn.edu.tw/ai-fml-international-academy/ieee-cis-education-portal.
OpenAI (2024) Transforming work and creativity with AI. https://openai.com.
Rahulnyk (2024), Convert any corpus of text into a graph of knowledge. https://github.com/rahulnyk/knowledge_graph.
Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2022) High-resolution image synthesis with latent diffusion models. https://arxiv.org/abs/2112.10752.
Wang Z, Huang Y, Song D, Ma L, Zhang T (2024) PromptCharm: Text-to-Image generation through multi-modal prompting and refinement. https://arxiv.org/abs/2403.04014.
Science & Technology Policy Research and Information Center (NARLabs) (2024) Trustworthy AI Dialogue Engine (TAIDE). https://taide.tw.
Kim M, Orshulevich V, Vardanian A (2024) UForm by Unum Cloud. https://huggingface.co/unum-cloud.
Espinoza-Guerra V (2024) Protogen Infinity Official Release. https://huggingface.co/darkstorm2150/Protogen_Infinity_Official_Release.