The Relationship between Music and AI/QML

Maiol Sabater Oliveras

February 4, 2025

Abstract– Music and quantum computing, though seemingly disparate fields, share underlying principles of complexity, patterns, and structure. This paper explores the intersection of these domains, asking two key questions: What can music offer to quantum computing? and What can quantum computing offer to music? By investigating these questions, this study aims to uncover how the rhythmic and harmonic richness of music might inspire quantum computational techniques and, conversely, how the unique properties of quantum systems might provide new tools for musical composition and analysis.

Keywords– Quantum Computing, Music Composition, Artificial Intelligence, Quantum Algorithms, Harmonic Patterns, Rhythmic Complexity, Superposition, Entanglement, Interdisciplinary Research, Generative Systems, Quantum Machine Learning, Musical Analysis, Computational Creativity

1 INTRODUCTION

MUSIC and quantum computing are two disciplines that, at first glance, appear to belong to vastly different realms—one rooted in human creativity and emotional expression, the other in the abstract mathematics and physics governing the quantum world [17]. Yet, both domains are deeply intertwined with the concepts of patterns, complexity, and structure. Music, through its rich interplay of rhythm, melody, and harmony, mirrors the elegant complexity found in quantum systems, where particles interact in probabilistic and entangled ways. This shared foundation presents a unique opportunity for interdisciplinary exploration.

The relationship between music and technology has a long history, with advances in fields such as acoustics, digital signal processing, and artificial intelligence profoundly shaping musical creation and analysis. Quantum computing, as a nascent but rapidly evolving field, offers a novel paradigm of computation that leverages the principles of superposition, entanglement, and interference [17]. These principles have the potential to revolutionize not only computation but also areas of human creativity, such as music [15]. Similarly, music's intricate structures and its historical role in inspiring mathematical thought may provide innovative perspectives on quantum algorithms and system design.

Classical machine learning and deep learning techniques have significantly contributed to advancements in music-related research [8, 9]. A primary focus in this domain is the generation of music using these techniques. Consequently, most models associated with music generation belong to a category known as generative models. These models aim to produce novel outputs—such as new songs or tracks—based on given inputs. While the specific nature of these inputs may vary, the commonality across applications lies in leveraging machine and deep learning techniques to process and construct the generated outputs.

In the quantum computing field, research on music has also centered around the development of generative models [15]. However, quantum generative models differ fundamentally from their classical counterparts. Establishing relationships between classical and quantum models is essential to designing quantum generative approaches, which makes these models distinct and unique. For instance, a classical convolutional neural network (CNN) operates differently from a quantum convolutional neural network (QCNN) [6], highlighting the unique characteristics and behavior of quantum models.

This paper explores the dynamic interplay between these two domains, focusing on two central questions: *What can quantum computing/AI offer to music?* and *What can music offer to quantum computing/AI?* By examining these questions, we aim to uncover how musical principles might inspire advancements in quantum computational methods and how quantum computing could, in turn, enrich the processes of musical composition, performance, and analysis. To en-

• E-mail : sabater.maiol@gmail.com
• Final Year Project. Degree on Artificial Intelligence. UAB
• Tutors: Matias Bilkis and Fernando Vilariño. Computer Science Department, UAB & Computer Vision Center
• Cours 2024/25

sure accessibility for readers unfamiliar with quantum computing, Appendix A.1 offers an introductory overview of the field, providing the necessary background to follow the discussion.

The motivation for this project arises from the unique potential of emerging technologies—specifically quantum computing—to push the boundaries of what is possible in this domain. Coupled with a passion for artificial intelligence and music, this work aims to contribute meaningfully to the growing body of research at the intersection of these fields. By addressing key challenges and exploring existing paradigms, the goal is to advance our understanding and capabilities in quantum music generation while fostering interdisciplinary collaboration in this evolving area.

The structure of this paper is as follows: Section 2 outlines the objectives of the research, setting the foundation for the study. Section 3 presents the state of the art concerning the two central questions posed earlier. Section 3.1 adopts a philosophical perspective, examining the limits and potential of quantum computing in music. Section 3.2 takes a more technical approach, exploring how quantum computing benefits from music, with a detailed example—Quiko—provided in Section 3.2.1 [19]. Section 4 describes the step-by-step workings of Quiko, along with the experimental setup used to evaluate its performance. Section 5 reports the results of the experiment, covering both qualitative and quantitative findings. Section 6 discusses these results in depth, addressing both the philosophical implications outlined in the state-of-the-art review and the technical outcomes of the experiment. Finally, Section 7 presents a concise summary of the conclusions in a bullet-point format, highlighting the key contributions of the work.

2 Objectives

This paper can be dissected in several objectives:

1. Provide an initial approach to the quantum domain and investigate its impact on music generation.
2. Implement a previously developed quantum system, using it to conduct experiments and derive meaningful and explanatory conclusions about how these models influence the relationship between music-AI.
3. Discuss and demonstrate how music can serve as a medium for learning quantum concepts.
4. Reflect on how quantum systems, including the one implemented, shape and influence musical creation.
5. Contribute to the understanding of how quantum computing impacts musical creativity through both technical exploration and philosophical discussion.

3 State of the Art

3.1 What can quantum computing / AI offer to music?

To explore the first question—*What can quantum computing offer to music?*—which is inherently exploratory and philosophical, because it is about what music can gain. The concept of music is an artistic concept, a human-based concept in which, without the human factor, music itself has no sense. For example, why would an AI compose music if it is not for a human to listen to it? Therefore, because it is rather an exploratory and philosophical question, an interview was conducted with Scott Oshiro[1]. Oshiro is a flutist, electronic musician, music researcher, and technologist, as well as a PhD student at the Center for Computer Research in Music & Acoustics (CCRMA) at Stanford University. He is also the creator of Quiko, a concept that will be discussed in detail later in this paper. His unique expertise at the intersection of music and technology provides valuable insights into the potential synergies between quantum computing and music.

In the interview, several compelling aspects related to the first question were explored. According to Eduardo Reck Miranda in his book *Quantum Computer Music: Foundations, Methods, and Advanced Concepts* [15], the author introduces three levels of abstraction in music:

- The microscopic level, focusing on frequencies and their arrangement.
- The note level, a more abstract layer dealing with musical notes.
- The building-block level, concerning the structural elements of composition.

These levels of abstraction suggest distinct ways to approach music composition, ranging from the granular manipulation of frequencies to more holistic structures. Building on this framework, the following question was posed to Scott Oshiro: *Is building a model to make music a new paradigm? For example, could there be a fourth level of abstraction called meta-composition, where the composer does not interact with intrinsic music at all?* Oshiro responded affirmatively, stating: *"Yes, you could call it whatever you want—if you like to call it meta-composition, so be it. The key idea is that the composer becomes the programmer of the model, rather than its end user."* This perspective highlights a shift in the composer's role, emphasizing the creation of generative systems over direct engagement with musical material.

Another question addressed the potential contributions of quantum machine learning (QML) to music: *"In what sense do you think AI can contribute to music—new structures, sounds, tunes, or genres?"* Oshiro replied: *"AI cannot explore beyond what it has seen. Ultimately, it composes based on the data it was trained on. It will not invent new methods or sounds but will create music grounded in its existing dataset."* This view aligns with that of Xavi Serra, a professor and director at the Music Technology Group at Universitat Pompeu Fabra. During the *Jornades sobre Música i IA* (a symposium on Music and AI), Serra remarked: *"Since technology adapts to the known realm, artists are forced to search in the unknown."* [22] Both perspectives underscore the limitations of AI in generating genuinely novel musical ideas, positioning the artist as the critical agent for exploring uncharted sounds, rhythms, and

[1] Interview with Scott Oshiro, PhD student at CCRMA, Stanford University.

structures. While AI provides tools for enhancing and organizing musical creation, the exploration of the unfamiliar remains a uniquely human endeavor.

The conceptual landscape of AI's capabilities can be divided into two distinct spaces: the explored space and the unknown space. The explored space refers to domains where data already exists, such as the variety of established music genres. In contrast, the unknown space pertains to areas where no data currently exists—for instance, entirely new genres of music that humanity has yet to conceive or create. Current research suggests that AI models are capable of navigating the explored space, but they struggle to address the unknown space effectively.

Two critical problems in AI exemplify this limitation: the Open Set Problem [25] and the Open World Problem [1], presented in the seminal works of Scheirer et al. and Bendale et al., respectively. The Open Set Problem addresses the challenge of handling situations where an AI encounters data or classes during testing that were not present in its training dataset. Conversely, the Open World Problem extends this challenge by requiring AI systems to operate in dynamic, unpredictable environments, continually updating their knowledge and adapting to new inputs. Unlike the Open Set Problem, which focuses on identifying and managing unknown inputs, the Open World Problem necessitates learning and functional adaptation over time. Various solutions to these problems have been proposed, including techniques based on Machine Learning (ML) [26], Reinforcement Learning (RL) [21] and emerging approaches like Quantum Learning (QL) [13]. While these methods provide insights into how AI systems can adapt to new information, they do not entirely address the issue of exploring the unknown space as defined earlier. At first glance, these three problems—the explored/unknown space issue, the Open Set Problem, and the Open World Problem—may seem similar. Each involves the division of a domain into known and unknown spaces, with the shared objective of learning and adapting to the unknown. However, the key distinction lies in the nature of the unknown space.

In the Open Set and Open World problems, the unknown space is only unknown to the AI system; it remains known to humans. For example, a system encountering a new class of data may find it unfamiliar, but that class exists within human knowledge. By contrast, in the case of exploring new music genres, the unknown space is unknown to both the AI system and humanity. There is no existing data or conceptual framework for such genres, making the challenge fundamentally different.

This distinction is further highlighted by phenomena such as model collapse [3], where generative models trained solely on their predecessors' outputs produce increasingly inaccurate results. Indicating that recursive training can degrade a model's ability to generate meaningful content. Additionally, studies like "Can LLMs Generate Novel Research Ideas?" by Chenglei Si et al. [5] show that while AI models can assist in idea generation, they often lack the originality and depth characteristic of human creativity. These findings underscore the difficulty AI faces in creating data that humans have not yet imagined.

All these studies and concepts provide valuable insights into whether AI can infer entirely new music genres and further consolidate the problem of exploring unknown spaces in AI. This issue remains a frontier challenge, distinct from the Open Set and Open World problems, as it involves pushing the boundaries of creativity and innovation into territories that are uncharted even by human cognition.

Another subject to consider is how AI tools contribute to music as a concept of art. Xavi Serra, in the same symposium mentioned earlier, affirmed that Spotify already hosts millions of songs generated by AI, many of which have never been listened to by a human. These AI-generated pieces often saturate Spotify with generic, overly simplistic music that, according to many experts, would not be considered art. On this topic, Oshiro was asked: *"Do you think music will lose its value as art because of AI?"* To which he responded: *"At the end, yes, I don't see why that would not happen."*

Oshiro's perspective aligns with Xavi Serra's view on how AI will affect the concept of art as it applies to music. While these algorithms may democratize music, making it accessible to a broader audience, both experts agree that the intrinsic artistic value of music could be diminished in the process.

3.2 What can music offer to quantum computing/Ai?

The previous question had a more philosophical nature, focusing on abstract considerations. In contrast, this question delves into what quantum computing and AI can directly gain from music. Unlike philosophical inquiries, these domains are not inherently human-centric, relying instead on measurable metrics and evaluations for their exploration and validation. This distinction opens the door to experimental approaches.

Before presenting the experiment, it is valuable to consider the question: *What can music offer to quantum computing and AI?* One perspective is that music can serve as a framework or metaphor for exploring and explaining quantum phenomena. Music, with its intricate patterns and inherent complexity, provides a conceptual tool that can aid in understanding and advancing quantum science for example Miranda's book [15].

Significant progress has already been made in the application of artificial intelligence (AI) to music composition. Early work dates back to 1957, when Lejaren Hiller and Leonard Isaacson composed *The Illiac Suite* using an early computer, guided by a set of music rules [10]. Later, in 2002, the music research team at Sony Computer Science Laboratory Paris developed the Continuator, an AI algorithm capable of continuing an artist's finished piece [20]. By 2010, Iamus became the first AI to generate a fragment of original contemporary classical music in its own distinctive style, known as "Iamus' Opus 1" [11].

Notably, Recurrent Neural Networks (RNNs) were initially devised for signal processing—which naturally includes music. One such example is a study from Stanford, *Music Composition Using Recurrent Neural Networks*, where a GAN architecture incorporated an RNN to generate music [23]. More recently, state-of-the-art models have grown even more sophisticated, integrating lyric creation. Examples of these advancements include Suno AI, launched in December 2023 [2], and Udio, released in April 2024 [24]. As demonstrated in the aforementioned examples, music and signal processing have the potential to inspire significant advancements in artificial intelligence (AI). These domains not only provide complex, structured data but also present challenges that drive the development of novel computational approaches. For instance, the field of signal processing, with its focus on analyzing and interpreting temporal sequences, has been a foundational catalyst for the creation and exploration of recurrent neural networks (RNNs). This relationship highlights how music and related signal processing tasks can shape and advance AI methodologies, fostering discoveries that extend beyond their initial domains.

Parallel to these developments in classical AI, the quantum realm has also seen promising ventures. Q1Synth [7], devised by Eduardo R. Miranda and collaborators, stands out as an innovative musical instrument producing sounds derived from quantum state vectors—reflecting qubit properties and corresponding measurements. In *Quantum Computer Music: Foundations, Methods, and Advanced Concepts*, Miranda further explores various implementations bridging quantum science and music, illustrating how musical structures can serve as a medium to interpret and model quantum phenomena [15]. Another notable different approach is a work done by Paulo Vitor Itaboraí [12] where the study investigates using the Variational Quantum Eigensolver (VQE) algorithm for generating chord progressions and sonifying the optimization process in music composition. Other research has focused on generating music using alternative quantum approaches—such as “Music Composition Using Quantum Annealing” by Ashish Arya, Ludmila Botelho, Fabiola Cañete, Dhruvi Kapadia, and Özlem Salehi [4] —which differs from the gate-based approach this paper will explore. Other steps towards different approaches have been taken, for example Eduardo Rick Miranda combined NLP and music in his work A Quantum Natural Language Processing Approach to Musical Intelligence [16]. As seen many different approaches can be taken to solve one single task, generate new music.

Although both classical and quantum models share the label “generative”, they operate on fundamentally different principles. Classical approaches rely on neural networks and backpropagation, whereas quantum approaches utilize circuits and gates [14, 6]. Consequently, even though both methods aim to tackle generative tasks, the strategies they employ to achieve this objective diverge significantly, one relies on a network-neuron structure and the other on quantum circuits.

These approaches make use of music to explore quantum mechanics by offering intuitive representations of complex concepts. By leveraging the structured and creative principles of music, researchers may uncover new pathways for advancing quantum computing and AI, using music not only as a medium of expression but also as a tool for scientific discovery.

To test this hypothesis—whether music can indeed aid in explaining and understanding quantum phenomena—this project incorporates Quiko, a quantum music generation application [19].

4 QUIKO

QuiKo [19] is an innovative quantum music generation application designed by Scott Oshiro, which appears in the book named Quantum Computer Music Foundations, Methods, and Advanced Concepts by Eduardo Reck Miranda (reference). It merges Quantum Machine Learning (QML) encoding techniques with a variety of quantum algorithms to produce unique drum samples. The system draws upon a database of audio tracks, using advanced quantum methodologies to analyze and reinterpret musical patterns. The core concept behind QuiKo lies in its ability to extract both harmonic and percussive elements from an input audio track. These extracted features are then encoded into the quantum circuit of QuiKo, allowing the quantum system to process this musical information. Through precise quantum measurements, QuiKo generates a probability distribution that serves as the foundation for constructing new beats. This approach enables the system to introduce an element of randomness and flexibility, leveraging the intrinsic properties of quantum mechanics. The idea of using quantum computing in addition to traditional computations in the process is to allow a more flexible and organic composition taking advantage of the noise produced by a quantum device.

More specifically Quiko uses two types of encodings, Static encoding and PKBSE encoding. Both rely on the concept of QPE in order to calculate the phase of the circuit and extract the necessary features for the similarity calculation. Altough only the PKBSE encoding has been implemented due to being the most efficient our of the two encodnign methods mentioned.

The whole project has used IBM’s quantum framework Qiskit. As Oshiro refers in his chapter, Quiko can be divided in 3 different blocks: Pre-processing, QuantumCircuit, and Analysis or Beat construction, Figure 1.

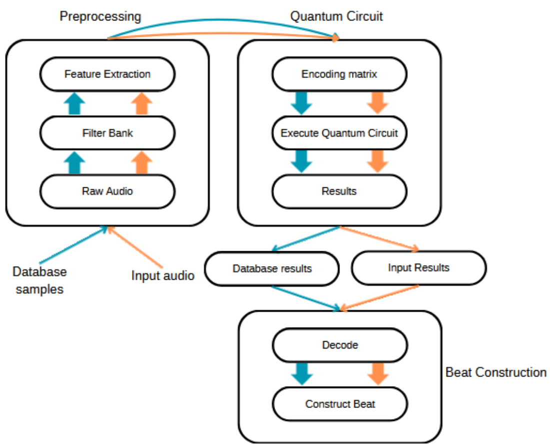

Fig. 1: Quiko Architecure

5 EXPERIMENTAL SET UP

5.1 Database Construction

The first step involves collecting a dataset of drum sound samples. Once the dataset was prepared, the samples were loaded using the librosa library. After loading a sample, three frequency subbands were created using filter banks. The filter banks have different bounderies in frequency depending on the number of qubits. Once the filter banks is applied to each of the dataste samples, it should yield in this case, since 3 qubits are being used, a low frequencies subband, a mid frequency and finally a high frequency subband for each sample. For each subband in each sample, harmonic and percussive features were extracted using the Harmonic-Percussive Source Separation (HPSS) algorithm. For the percussive component, the peaks (local maxima) in the signal were identified, and their values were summed to compute a final value. This value will later be used in the implementation of U gates as the value θ. For the harmonic component, the three highest peaks in the signal were identified, and a weighted average of these peaks was computed. This weighted average serves as the parameter ϕ for the U gate. Additionally, the spectral centroid of the harmonic component was calculated, which will be used as another parameter, λ, in the quantum U gate configuration. Equation (1) formalizes the computation performed by a U gate, while equations (2), (3), and (4) detail the calculations for the respective parameters: θ, ϕ, and λ, as previously explained.

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & e^{-i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{pmatrix} \tag{1}$$

$$\phi = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)} \tag{2}$$

$$\lambda = \max\{f(n)_{\text{onset}}\} \tag{3}$$

$$\theta = \arg\max_{x=s} \left\{ \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi kn}{N}} \right\}, \quad k = 0, \ldots, N-1 \tag{4}$$

The angles θ, ϕ, λ for the low-frequency subband are mapped to qubit 0. Similarly, the corresponding angles for the mid-frequency subband are mapped to qubit 1, and the same process is applied to the high-frequency subband for qubit 2. As a result, for each sample in the database, a quantum circuit is constructed following Figure 2 structure.

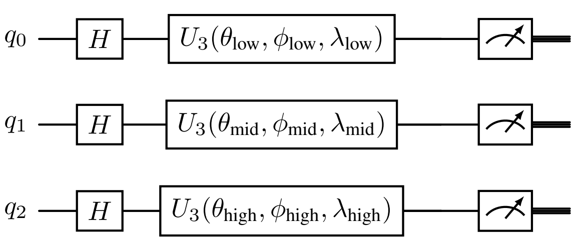

Fig. 2: Database Sample encoding

Some observations regarding this encoding are as follows: Any other parameter can be encoded as θ, ϕ, λ values, making the encoding flexible. While the method itself is relatively simple, it relies on classical pre-processing to generate the parameters for the quantum circuit. Additionally, because the resulting circuit is straightforward, the quantum aspect is not strictly necessary. In fact, the same computations could be performed classically with identical results. Finally note how there is no measure layer in the circuit, the idea for calculating the fidelity is to get a state vector since the code is using a simulator. However, this approach serves as a useful introductory example to demonstrate the principles of quantum encoding and familiarize users with the process.

5.2 Input sample processing

The process for encoding the input audio is almost analogous to the process used for the samples in the database. The input audio is filtered through the same filter bank to extract the three previously mentioned subbands: low, mid, and high. For each subband, the Harmonic-Percussive Source Separation (HPSS) algorithm is applied to isolate the harmonic and percussive features of the audio as show in Figure 3.

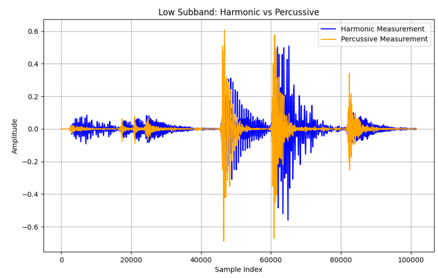

Fig. 3: Harmonic vs. Percussive elements in the low subband

The harmonic and percussive segments are then divided into $2^{\text{number of qubits of subdivision register}}$. The subdivision register essentially determines the duration of the beat before it repeats and processes rhythmic information, in this case this register will be used as estimation register, further explained later in the paper. If three qubits are allocated to the subdivision register, yielding $2^3 = 8$ subdivisions, meaning the beat will span 8 units.

The interpretation of these 8 subdivisions depends on the chosen rhythmic context. If the subdivisions are interpreted as crotchets (quarter notes), the beat lasts for two bars (or measures) in a $4/4$ time signature, where each bar consists of 4 crotchets. Conversely, if the subdivisions are interpreted as quavers (eighth notes), the beat spans only one bar. In general, the duration of the beat, in terms of measures, can be formalized as equation (5) shows.

$$\text{Measures} = \frac{2^{\text{Spinal Cord Qubits}}}{\text{Notes per measure}} \tag{5}$$

$$\text{Notes per measure} = \begin{cases} 2 & \text{for minim (half note)}, \\ 4 & \text{for crotchet (quarter note)}, \\ 8 & \text{for quaver (eighth note)}. \end{cases}$$

Note: These values apply to a 4/4 time signature.

So basically, the eight subdivisions in the harmonic and percussive segments are used to calculate, as before, the respective θ, ϕ, and λ values for the U gates of the Timbre register, the target register. For each subdivision, a U gate will be applied for each subband. So if there are 8 subdivisions, that means that for each subband, low, mid and high, there will be 8 U gates. On one side, there is the timbre register, which is composed of 3 qubits. The function of these qubits is the same as the 3 qubits in the database circuits: to encode the θ, ϕ, and λ for the 3 different subbands. On the other side, there is the subdivision register, which determines the duration of the beat before it repeats itself.

Both of these registers are utilized to construct the circuit illustrated in Figure 4. The encoding algorithm employed is Quantum Phase Estimation (QPE), where the subdivision register functions as the control register, and the timbre register serves as the target register for the algorithm. The core concept is that the control register encodes the audio signal into a quantum state, while the subdivision register encodes the phase. This phase represents the subdivision within which the timbre register's encoding corresponds to a musical measurement (bar). The encoding of the timbre register is subsequently compared with the database to identify the sample most closely matching the given phase.

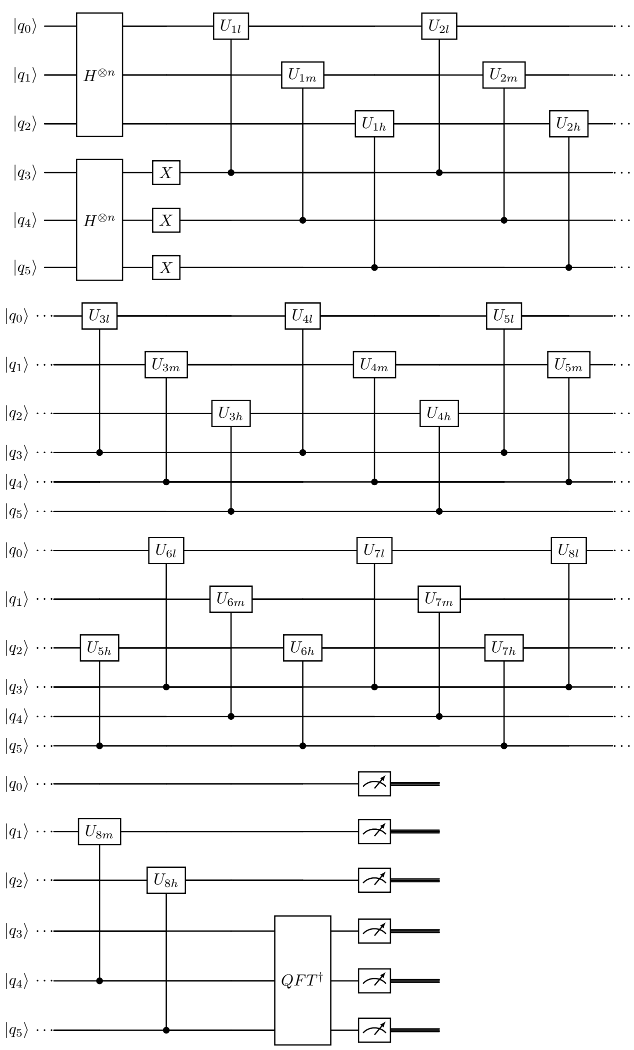

Fig. 4: Input Sample encoding

5.3 Measurements

In the original study, the experiment was repeated multiple times to construct a histogram representing the number of occurrences of the system collapsing into a specific state. From this histogram, the probabilities of each state were calculated. However, this approach relies on classical data derived from experimental results, meaning the probabilities are not inherently quantum. In contrast, since this work uses a simulator, an alternative approach can be adopted.

Rather than measuring the system directly, the simulator provides access to the exact probabilities of each state at the end of the circuit as mentioned in Equation 9 in Appendix A.1. Referring to Equation 6, the coefficients c_i represent the complex amplitudes in the superposition. By definition, the square modulus of these coefficients, $|c_i|^2$, represents the probability that a measurement of the observable will collapse the system to the eigenstate ψ_i.

$$\sum_i |c_i|^2 = 1. \tag{6}$$

This capability of getting the coefficients directly allows us to easily calculate the fidelity. The process involves running the given input sample through its circuit to obtain a state vector with the coefficients information. Subsequently, each sample in the database is executed through its respective circuit, generating as many state vectors as there are samples in the database.

It is important to emphasize that, due to the different numbers of qubits in the circuits, the state vector of the input sample will differ from the state vectors of the database samples. This distinction arises because the number of qubits determines the dimensionality of the state vector. The possible states of a circuit can be calculated by 2^n where n is the number of qubits.

The database samples will have eight possible states (three qubits), while the input sample will have sixty-four states (six qubits). The input sample states can be represented as $|000000\rangle$, $|000001\rangle$ ... $|111111\rangle$. In this context, the first three qubits of the six can be interpreted as representing the subdivision in the measure of the beat, the representation of the phase, while the remaining three qubits correspond to harmonic and percussive elements. For example, the state $|011011\rangle$ can be interpreted as representing the third subdivision (011) of the measurment and the percussive and harmonic features at the state 011.

Fig. 5: state $|010011\rangle$ corresponds to 3rd note of the measure

By this logic, the sixty-four states of the input signal can be categorized into eight subdivisions, each containing eight states, yielding $8 \cdot 8 = 64$. To compare two samples—specifically, the input signal and a sample from the database, compute the fidelity of the ith database sample $\psi_{\rho,i}$ with the jth subdivision of the input sample, $\psi_{\sigma,j}$, Equation 7.

$$F(\rho, \sigma)_{i,j} = |\langle \psi_{\rho,i} | \psi_{\sigma,j} \rangle|^2 \tag{7}$$

Once the fidelities for each sample in the database for each subdivision are computed, the fidelities can be sorted in order to get the ten most similar samples for each subdivision. Also since the coefficients of the input sample were extracted, the probabilities of each subdivision to appear can be calculated, therefore providing the ten most similar samples per subdivision along with the probability of that subdivision to sound.

Overall, the whole system can be viwed as Figure 6 which provides a visual representation of what Quiko is doing.

6 Results

To conduct the experiment, the 9th Wonder sound pack was selected to construct the database, as it was the original sound pack used in the referenced study. This hip-hop sound pack consists of seven categories of drum sounds: claps, cymbals, hi-hats, snares, shakers, percussion, and kicks, each containing multiple samples. For the input audio, the GTZAN dataset was employed, specifically the 100 samples categorized under the hip-hop genre, this due to both database and input audios being the same genre. While not all input samples were used to reconstruct new beats, they were analyzed to provide insights into the functionality and behavior of Quiko.

The experiment utilized three qubits to represent the database samples and six qubits for the input sample. The initial phase involved constructing quantum circuits for all database samples. Although the original code served as a strong foundation, it was adapted to meet the specific requirements of the experiment. To improve efficiency, the circuits were saved after their initial construction, eliminating the need for reconstruction during subsequent executions. Instead, the pre-constructed circuits were directly loaded, thereby streamlining the overall process.

Subsequently, the hip-hop samples were processed. Prior to their use in any circuit, the samples underwent preprocessing. Since the goal was to construct a musical measure, only one measure per sample was required. By utilizing the librosa beat_track function, the tempo of each sample was computed, allowing for the extraction of a single measure from the desired track. This extracted measure was then used as the input for the circuit architecture. Once the input and database circuits were prepared, they were executed, and the fidelity between the input and database samples was calculated. To refine the results, only the ten database samples with the highest similarity values were retained, along with the probability of each subdivision appearing. For the plots in Figures 7 and 8, the sample hiphop.00031 was used. These figures illustrate the probability distribution of each subdivision and the similarity of the three most similar samples for each subdivision.

As observed in this sample, the similarities and probabilities of the subdivisions are generally aligned. The subdivisions with the highest probabilities of being played also correspond to those with the most similar samples, with the exception of subdivision 2. Subdivision 2 exhibits a high probability of being played but demonstrates lower fidelity compared to subdivisions 0, 4, and 6. This result highlights how the most frequently played subdivisions—namely 0, 2, 4, and 6—tend to dominate, as these subdivisions align with the natural rhythmic grounding of the music.

This observation provides valuable insight into the independence of similarities and probabilities. Even if one sample is more similar to another, a sample from a more probable subdivision will have a higher likelihood of sounding, regardless of its similarity. This highlights a nuanced relationship between probability and similarity in this context.

In order to analyze how the subdivision probabilities are distributed across all subdivisions, the same process used for the hiphop.00031 sample was repeated with the other 99 samples to see if with different samples the result was the same. The probabilities were aggregated and averaged, resulting in Figure 9, which illustrates the probability distribution across subdivisions for these diverse examples.

As seen in Figure 9, the most probable subdivision are subdivisions 0,2,4 and 6 similar than the hiphop.00031 sample. This observation aligns with music theory, where the first beat of a measure is the most significant as it marks the beginning of the measure. Subdivisions 2,4 and 6 corresponding to the 1/4th, 2/4th and 3/4th of the measure respectively.

In this experiment, conducted using a 4/4 time signature, each measure is divided into eight subdivisions: beat 1 includes subdivisions 0-1, beat 2 includes subdivisions 2-3, beat 3 includes subdivisions 4-5, and beat 4 includes subdivisions 6-7. Beat 1, starting the measure, usually holds the highest importance (if the track is not upbeat), followed by beat 4, which marks the midpoint. The remaining beats complement these primary beats.

The representation shown in Figure 9 accurately reflects the expected probability distribution across subdivisions, showcasing Quiko's capability to effectively analyze rhythmic data. While the average probability distribution aligns well with music theory, not all individual sample probabilities are as evenly distributed. In fact, different distinct groups of probability distributions were identified, with the distribution type depicted in Figure 9 being the most prevalent. For further details on these distribution groups, refer to Appendix A.2.

Another important aspect to consider is the behavior of fidelity across all samples, as shown in Figure 10. In this figure, if a sample appears more than once within a subdivision, its similarity values are aggregated. Subdivision 0 contains the most similar samples, followed by subdivisions 2, 4, and 6, which corresponds to the probabilities of these subdivisions. Beyond these subdivisions, no clear pattern is observed, indicating that the drum database is well-suited for the sampled songs.

Another key aspect to analyze is Quiko's variety in selecting which sample should play. To explore this, we refer to Figure 11, which illustrates, across all 100 hip-hop samples, the frequency with which each database sample appears within the top 10 samples with the highest fidelity per subdivision. Notably, subdivisions 1 and 4 exhibit the greatest variety. Additionally, specific samples such as Str_Oh1 and Lo_Tom1 appear frequently across different subdivisions, likely due to the duration of these database samples.

Except for subdivisions 1 and 4, which demonstrate high variability, the most frequently repeated database samples in other subdivisions tend to be the longest samples in the database. Specifically, the six longest samples (Str_Oh1,

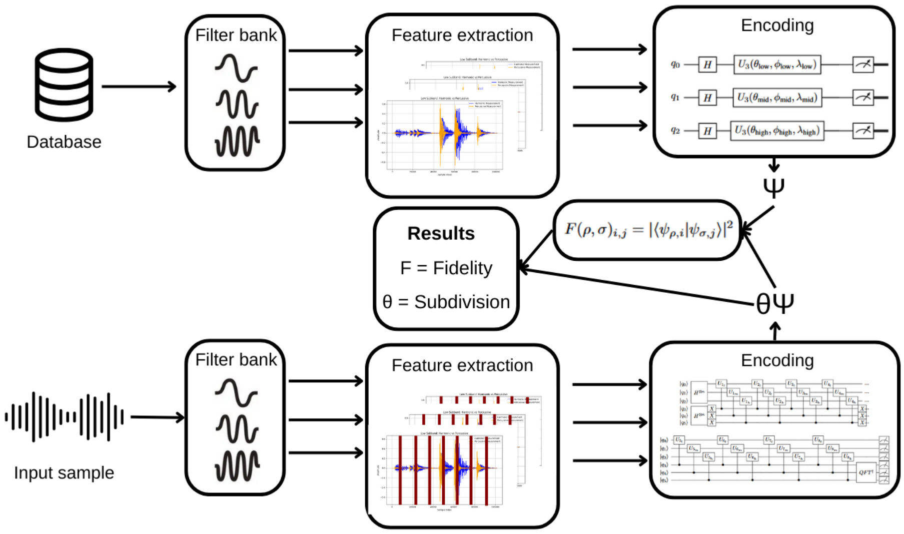

Fig. 6: Quiko Architecture

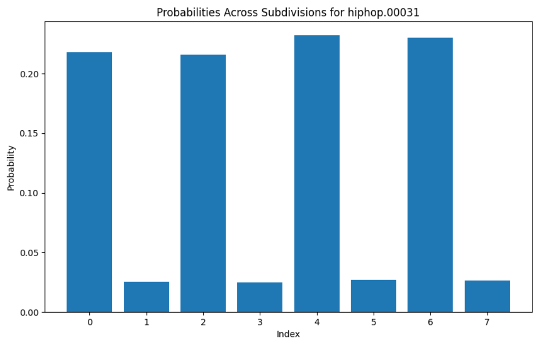

Fig. 7: Subdividsion probability over hiphop.00031 sample

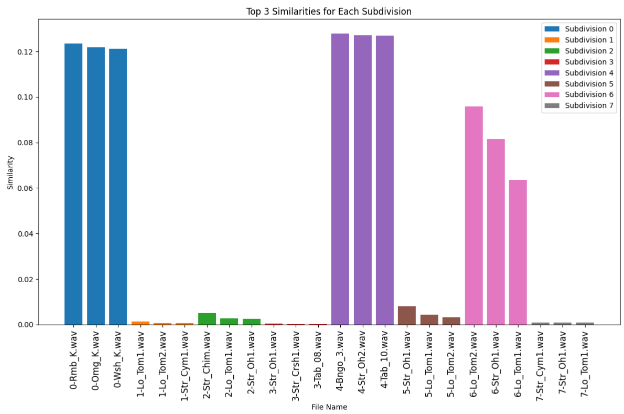

Fig. 8: Fidelity of samples per subdivision over hiphop.0031 sample

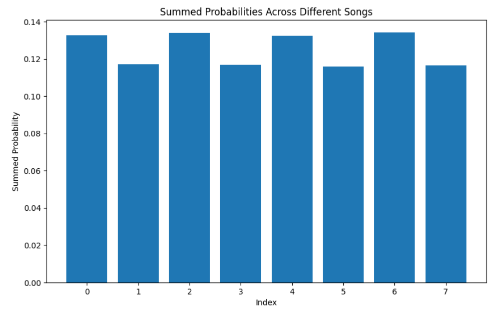

Fig. 9: Probability vs. Subdivision

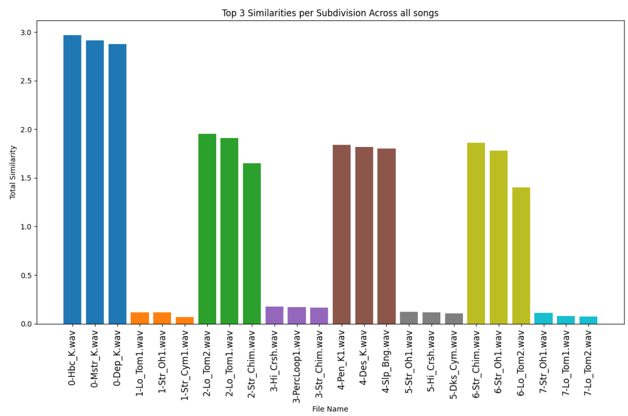

Fig. 10: Fidelity per samples per subdivision over all samples

Str_Chim, Str_Crsh, Lo_Tom1, Lo_Tom2, and Str_Cym1) dominate the outputs. This suggests that Quiko is biased toward longer samples, potentially due to how circuit values are computed during the analysis process.

Beyond rhythmic data analysis, this study also explores the generation of new drum beats for the samples used to calculate Figure 9. These generated beats can be accessed in this project's GitHub repository [18], that repository also contains a tutorial to execute the code. To create these beats, Quiko processes an input sample and identifies the top n most similar database samples for each subdivision, in this case 8 samples would be enough, along with their associated probabilities.

Using these probabilities as weights, a weighted random sampling is performed to determine the number of sounds allocated to each subdivision. Each new beat comprises a total of eight samples, a value chosen for its balance: fewer samples result in a sparse beat, while more samples create an overly cluttered rhythm within a short time frame. This

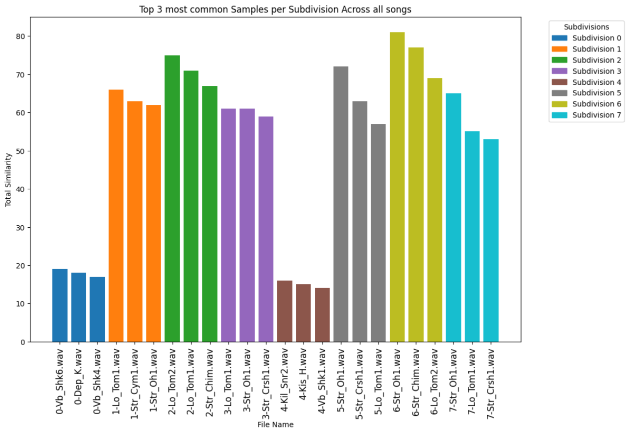

Fig. 11: Fidelity per samples per subdivision over all samples

approach ensures that the generated beats achieve an optimal rhythmic density.

7 Discussion

Given some interviews, participation in different meetings, and the conducted experiment on Quiko, this project addresses both questions:

1. What can quantum computing/AI offer to music?
2. What can music offer to quantum computing/AI?

On one hand, the first question is answered by the aforementioned interviews, previous investigation on the state of the art and the experiment itself. The interviewed experts align in their opinion that AI cannot easily produce data outside of the seen distribution. For music, this limitation implies that AI/QML will not explore new genres, sounds, or feelings with music—at least with current methods. This incapability of AI provides human musicians with a unique advantage: the ability to explore the unexplored. Therefore, an AI will never fully substitute a human musician since it cannot perform exactly what a musician does. Quiko exemplifies this by providing a tool where music benefits from AI/QML to create drum patterns. However, Quiko is limited by its training data: it can explore combinations within its database but cannot go beyond it. Also it has been proven that despite those limitations such as a database limitation or the fact that in the given database some subdivision don't have a lot of variaety, Quiko is capable of composing new beats for music.

This whole study with Quiko allows us to shade some light to the matter, according to the results we can conclude that on one side, Quiko is capable of generating new beats, these tools help musicians explore music, since it facilitates the process of making music, allowing the musicians to focuse more on exploring new genres and sounds rather than trying to explore the already existing sounds and generes. On the other side, Quiko's output is database dependent, altough Quiko is not near being a perfect tool for its function since it has been proved that quiko is a database dependent system and also suggesting a bias towards long samples in the database, it serves its porpuse of exploring the different combinations of samples in the database. Also mention that the utilization of circuits does not stop here, these tools ables a lot of utilities, in this case the feature extraction utility, where given an input it extracts some features to be compared with the others but many other applications can be drawn from these architectures.

Another nice conclusion it can be drawn from the experiment is that in terms of art, AI/QML has democratized music, enabling more people without specialized knowledge to start producing music, for instance a person without previous musical knowledge can use any of these tools such as Quiko, to generate and experiment with music. On the positive side, this democratization allows everyone to experiment with music. However, it also risks diminishing the artistic value of music because:

1. Not all music produced by AI is of high quality.
2. Art is a human concept.

Ultimately, AI/QML serve as powerful tools for music to develop and explore known elements, although potentially risking the artistic essence of music.

On the other hand, the second question arises. Using Quiko as an example again, music serves as a tool to introduce quantum concepts such as:

- Qubit
- Quantum Circuit
- Quantum Phase Estimation
- Quantum Probabilities
- Quantum Gates

Although only a few concepts are introduced in this paper, this demonstrates how quantum phenomena can be understood through music, provided the proper tools are available. Before starting this project, I had no prior knowledge of quantum computing. Yet, with a solid foundation in music and external assistance, I was able to understand and reproduce Quiko. This proves that music can elucidate quantum phenomena, leaving room for potential discoveries in the quantum field driven by musical applications.

Quiko serves as an excellent example due to its demonstrated ability to extract meaningful patterns from music. By analyzing a sample from a song and generating a new beat for it, Quiko identifies key features such as how a measure should be distributed.

In conclusion, Quiko is a useful and foundational tool that employs quantum machine learning (QML) at its core to compute new drum bases. Its strength lies in effectively encapsulating the probability distribution of a measure. However, the quality of the output is highly dependent on the database provided, making the choice of database critical to achieving satisfactory results.

8 Conclusions

- AI and Quantum Machine Learning (QML) have a symbiotic relationship with music.

- Regarding objective 1, AI and QML democratize music creation. Tools like Quiko exemplify this democratization by making music production more accessible. However, this increase in accessibility does not inherently result in higher-quality outputs but rather a greater quantity of music being produced.
- According to objective 2, Quiko demonstrates the potential of AI and QML in music by generating drum patterns based on input samples. This highlights how these technologies can contribute creatively within specific parameters.
- Regarding objective 3, music provides an intuitive framework for introducing and explaining quantum concepts, such as qubits, quantum circuits, quantum phase estimation, probabilities, and gates. The structured nature of music makes it an effective medium for communicating these complex ideas.
- Thanks to the exploration of Quiko, we can address objective 4, concluding that AI and QML cannot generate musical data outside their training distribution. This limitation restricts their ability to innovate in new genres, sounds, and emotional expressions. Quiko, for example, depends on existing datasets to create new outputs, highlighting this dependency.
- For objective 5, music-inspired quantum tools like Quiko demonstrate the potential to advance both quantum computing and music. By bridging artistic and scientific paradigms, such tools foster discoveries that integrate creative and technical domains.

9 Future Work

- Train the parameters in the quantum circuits to improve performance.
- Regularize the audio to mitigate the bias toward longer samples, as previously mentioned.
- Formalize a systematic method to select the parameters for each track, rather than relying on arbitrary choices.
- Create a more diverse and balanced database with a wider variety of audio samples.
- Automate the process of reconstructing the beat.

Acknowledgements

I am deeply grateful to Matias Bilkis and Fernando Vilariño for their direct assistance with this project, dedicating numerous hours to improving and refining this work as I embarked on my first steps in the quantum realm. I also wish to thank the CVC and its QML group for providing the support, resources, and opportunity to carry out this research as part of my internship. Lastly, I extend heartfelt appreciation to my mother, whose early encouragement towards music when I was a kid eventually led me to pursue this project.

References

[1] Terrance Boult Abhijit Bendale. Towards open world recognition. arXiv preprint arXiv:1412.5687, 2014.

[2] Suno AI. Suno ai: Integrating lyrics and music generation. Launched in December 2023, 2023.

[3] Alexandra Jonker Alice Gomstyn. What is model collapse?, October 2024. Accessed: 2025-01-12.

[4] Ashish Arya, Ludmila Botelho, Fabiola Cañete, Dhruvi Kapadia, and Özlem Salehi. Music composition using quantum annealing. arXiv preprint arXiv:2201.10557, 2022.

[5] Tatsunori Hashimoto Chenglei Si, Diyi Yang. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. arXiv preprint arXiv:2409.04109, 2024.

[6] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. arXiv, 2019.

[7] Peter Thomas Eduardo Reck Miranda and Paulo Vitor Itaboraí. Q1synth: A quantum computer musical instrument. https://doi.org/10.3390/app13042386, 2023.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[9] David Ha, Andrew M. Dai, and Quoc V. Le. Neural audio synthesis of musical notes with wavenet autoencoders. *arXiv preprint arXiv:1704.01279*, 2017.

[10] Lejaren Hiller and Leonard M. Isaacson. The illiac suite: A landmark computer-generated composition. Composition, 1957. One of the earliest works of computer-assisted composition.

[11] Iamus. Iamus' opus 1. Computer-Generated Composition, 2010. First AI to generate a fragment of original contemporary classical music in its own style.

[12] Paulo Vitor Itaboraí, Tim Schwägerl, María Aguado Yáñez, Arianna Crippa, Eduardo Reck Miranda Karl Jansen, and Peter Thomas. Variational quantum harmonizer: Generating chord progressions and other sonification methods with the vqe algorithm. arXiv preprint arXiv:2309.12254, 2023.

[13] Changlin Song Jin Wang. Quantum learning and essential cognition under the traction of meta-characteristics in an open world. arXiv preprint arXiv:2311.13335, 2023.

[14] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. arXiv, 2018.

[15] Eduardo R. Miranda. *Quantum Computer Music: Foundations, Methods, and Advanced Concepts*. Springer, Cham, Switzerland, 2022.

[16] Eduardo Reck Miranda, Richie Yeung, Konstantinos Meichanetzidis Anna Pearson, and Bob Coecke. A quantum natural language processing approach to musical intelligence. arXiv preprint arXiv:2111.06741, 2021.

[17] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[18] Maiol Sabater Oliveras. Quantum music, 2025. https://github.com/MaiolSabater/Quantum-Music.

[19] Scott Oshiro. Quiko: Quantum beat generation application. *Arxiv*, 2022. Proposes a quantum music generation application.

[20] François Pachet. The continuator project. Sony Computer Science Laboratory Paris, 2002. AI algorithm to continue an artist's unfinished piece.

[21] Ashley Peake, Joe McCalmon, Yixin Zhang, Sarra Alqahtani Daniel Myers, and Paul Pauca. Deep reinforcement learning for adaptive exploration of unknown environments. arXiv preprint arXiv:2105.01606, 2021.

[22] Xavi Serra. Remarks at jornades sobre música i ia (symposium on music and ai). Symposium presentation, Universitat Pompeu Fabra, 2024.

[23] Jane Smith and Alex Brown. Music composition using recurrent neural networks. Stanford University Report, 2022. Exploring GANs with RNNs for music generation.

[24] Udio. Udio: Ai music generation platform. Released in April 2024, 2024.

[25] A. Sapkota W. J. Scheirer, A. de Rezende Rocha and T. E. Boult. Toward open set recognition. IEEE transactions on pattern analysis and machine intelligence, pp. 1757–1772,, 2012.

[26] Peng Zhao, Jia-Wei Shan, Yu-Jie Zhang, and Zhi-Hua Zhou. Exploratory machine learning with unknown unknowns. arXiv preprint arXiv:2002.01605, 2020.

APPENDIX

A.1 Introduction to quantum computing

A.1.1 Qubit

In classical computing, information is encoded in bits, which can take on one of two distinct values: 0 or 1. In quantum computing, the analogous unit of information is the qubit, which operates on fundamentally different principles. Unlike classical bits, which exist exclusively in one state (0 or 1) at any given time, qubits exhibit a property known as **superposition**. This property enables a qubit to exist in a linear combination of both states simultaneously, mathematically represented as equation 8.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{8}$$

where:

- α: A complex number (coefficient) representing the amplitude of the state $|0\rangle$
- β: A complex number (coefficient) representing the amplitude of the state $|1\rangle$

To better understand what α and β are, qubtis have a property expressed in equation 9 where $|\alpha|^2$ and $|\beta|^2$ can be understood to the probabilities of the states $|0\rangle$ and $|1\rangle$ to appear upon measurement.

$$|\alpha|^2 + |\beta|^2 = 1 \tag{9}$$

To illustrate these concepts with an example, consider a qubit in a state where both $|0\rangle$ and $|1\rangle$ are present, with squared coefficients $|\alpha|^2 = 0.5$ and $|\beta|^2 = 0.5$. In this case, the qubit is in **equal superposition** because both states have the same probability of appearing upon measurement. Before measurement, each state has a probability of 50% to appear. However, upon measurement, the system will **collapse** into one of the two states. For instance, if the system collapses into the state $|1\rangle$, then $|\alpha|^2 = 0$ and $|\beta|^2 = 1$.

A.1.2 Bloch Sphere

To visually represent a qubit, we use the Bloch Sphere, a unitary sphere depicted in Figure 12, where the state $|0\rangle$ is positioned at the top of the sphere and $|1\rangle$ at the bottom.

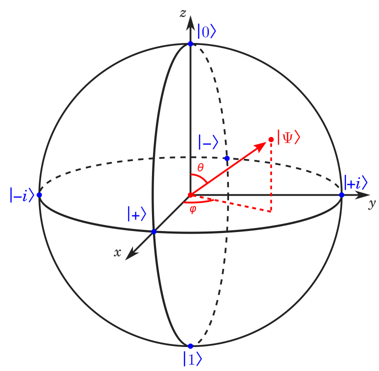

Fig. 12: Representation of a qubit into a unitary sphere, Bloch Sphere

This representation suggests that the state $|\psi\rangle$ can be visualized as a point on the surface of the sphere, as described by Equation 10. While the Bloch Sphere effectively represents operations performed on a single qubit, it becomes insufficient for visualizing entangled qubits, where the complexity of interactions extends beyond this simple representation.

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{10}$$

A.1.3 Gates

Given a qubit, various operations can be performed on it using **quantum gates**, which are transformations applied to a single qubit or a set of qubits. For instance, consider the X gate, the most simple quantum gate, often referred to as the quantum equivalent of the classical NOT gate. In classical computation, the NOT gate flips the input: if the input is 0, it outputs 1, and vice versa. Similarly, the X gate in quantum computation performs this operation on a qubit. X can be defined as Equation 11

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{11}$$

Then referring to Equation 8, that operation of a qubit can be rewritten as shown in Equation 12.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{12}$$

In summary, applying the X gate to a qubit flips its state. Specifically, the X gate transforms the state $|0\rangle$ into $|1\rangle$, and $|1\rangle$ into $|0\rangle$, effectively interchanging the two basis states. Mathematically it looks like Equations 13, 14, 15.

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \equiv |1\rangle \tag{13}$$

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \equiv |0\rangle \tag{14}$$

$$X|\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \beta|0\rangle + \alpha|1\rangle \tag{15}$$

Therefore, when analyzing the X gate in terms of probabilities, it becomes evident that the gate swaps the coefficients α and β. Consequently, the probabilities associated with each state are also interchanged.
In addition to the X gate, there are several other quantum gates, each with unique functionality and mathematical representation. One of the most significant is the **Hadamard gate**, or H gate, which is primarily used to place a qubit into a state of superposition. The action of the Hadamard gate on a qubit is mathematically described by Equations 17, 18 and 19.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{16}$$

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \equiv |+\rangle \tag{17}$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \equiv |-\rangle \tag{18}$$

$$\begin{aligned} H|\psi\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} \alpha + \beta \\ \alpha - \beta \end{bmatrix} \\ &= \frac{\alpha + \beta}{\sqrt{2}}|0\rangle + \frac{\alpha - \beta}{\sqrt{2}}|1\rangle \end{aligned} \tag{19}$$

Equations 17 and 18 illustrate the effect of the H gate on the computational basis states. Using as reference Figure 12, the application of the H gate transforms the state $|0\rangle$ into $|+\rangle$ and the state $|1\rangle$ into $|-\rangle$, placing both states into equal superposition.

The H gate is often applied at the very beginning of a quantum algorithm. During initialization, the qubit is typically prepared in the ground state, $|0\rangle$, which requires setting the coefficients $\alpha = 1$ and $\beta = 0$. After the H gate is applied, the coefficients of the resulting state are given by:

$$c_0 = \frac{\alpha + \beta}{\sqrt{2}} = \frac{1 + 0}{\sqrt{2}} = \frac{1}{\sqrt{2}},$$
$$c_1 = \frac{\alpha - \beta}{\sqrt{2}} = \frac{1 - 0}{\sqrt{2}} = \frac{1}{\sqrt{2}}.$$

Squaring these coefficients, c_0^2 and c_1^2, yields probabilities of 0.5 for each state, corresponding to a 50% chance of observing either state. This demonstrates the superposition created by the Hadamard gate.
Both H and X gates are examples of **Unitary gates** or U gates. They are gates that performs a reversible transformation on a quantum state. In mathematical terms, a unitary gate is represented by a unitary matrix U which satisfies Equation 20. The idea is that every unitary matrix U has an inverse and the inverse is given by its conjugate transpose, Equation 21.

$$U^\dagger U = UU^\dagger = I \tag{20}$$

$$U^{-1} = U^\dagger \tag{21}$$

Another property is that the norm of a quantum state is preserved under unitary operations. This means that the Equation 9 will always stand. And finally its third property is that unitary gates correspond to linear transformations in the Hilbert space, meaning that the output of the gate applied to a combination of states is the combination of the outputs of the gate applied to each individual state, Equation 22.

$$U|\psi\rangle = U(\alpha|0\rangle + \beta|1\rangle) = \alpha U|0\rangle + \beta U|1\rangle \tag{22}$$

The general form of a U gate for a single qubit, represented as a unitary matrix, can be expressed as shown in Equation 23.

$$U = e^{i\lambda} \begin{bmatrix} \cos\theta & -e^{-i\phi}\sin\theta \\ e^{i\phi}\sin\theta & \cos\theta \end{bmatrix} \tag{23}$$

A.1.4 Quantum Phase Estimation (QPE)

Quantum Phase Estimation (QPE) is one of the most important tools in quantum computing. Its primary objective is to determine the eigenvalue of a unitary operator given one of its eigenstates.

To formalize the problem, consider a U gate and one of its eigenstates $|\psi\rangle$. The action of U can be expressed as shown in Equation 24:

$$U|\psi\rangle = e^{i\phi}|\psi\rangle, \tag{24}$$

where:

- ϕ represents the phase of the eigenvalue, or equivalently $\theta = \phi/2\pi$
- $e^{i\theta}$ is the eigenvalue

The goal of QPE is to estimate ϕ or θ given the operator U and the eigenstate $|\psi\rangle$.

The process involves constructing a quantum circuit, a sequence of operations, such that the transformation in Equation 25 holds. To extract the phase ϕ, only the last register of the circuit needs to be measured. A register refers to a collection of qubits treated as a single unit for computations. In this scenario, two registers are required: the **control (or estimation) register**, which encodes the estimated phase ϕ, and the **target register**, which holds the eigenstate of the unitary operator U.

$$|\psi\rangle|0\rangle \to |\psi\rangle|\phi\rangle. \tag{25}$$

Since the imaginary exponential has a periodicity of 2π, ϕ can be defined as $\phi = 2\pi\theta$, so from now on the phase will be refered as θ. The idea is that θ will be represented in the quantum computer as a binary number. For example, the value 0.3125 can be represented in binary as 0.0101:

$$0.0101 = 0 * 2^{-1} + 1 * 2^{-2} + 0 * 2^{-3} + 1 * 2^{-4}$$

$$0.3125 = 1 * 2^{-2} + 1 * 2^{-4}$$

This binary representations are interesting because qubits can be encoded into the value: $|0101\rangle = 0.0101$. As one may already have noticed, precision in this measurment depends on the number of qubits, the more qubits the more accurate since there would be more numbers to represent the measurment.

With the concepts explained, let us now look at how the Quantum Phase Estimation (QPE) algorithm works. At the beginning of the QPE process, the control register is initialized in the state $|0\rangle$, and the target register is initialized in the eigenstate $|\psi\rangle$, as shown in Equation 26:

$$|0\rangle \otimes |\psi\rangle. \tag{26}$$

Next, the control register, which is responsible for encoding the phase θ, is transformed into a uniform superposition state. This is achieved by applying Hadamard gates to each qubit in the control register. The resulting state ensures that all computational basis states $|k\rangle$ (where $k = 0, 1, \ldots, 2^{n-1}$) are equally weighted. After this transformation, the state can be described mathematically as shown in Equation 27.

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes |\psi\rangle. \tag{27}$$

Controlled versions of the unitary operator U are then applied to the system. For each control qubit j, the unitary U is raised to a power of $k = 2^j$, where j denotes the position of the control qubit. These controlled gates entangle the control register with the target register (which holds the eigenstate $|\psi\rangle$), encoding the phase information θ into the control qubits. This process results in the quantum state described in Equation 28.

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} U^k|k\rangle \otimes |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i2\pi k\theta}|k\rangle \otimes |\psi\rangle. \tag{28}$$

After these operations, the inverse Quantum Fourier Transform (QFT) is applied to the control register. This operation consolidates the phase information, which was previously scattered across the qubits, into a form where the binary representation of the phase θ can be directly measured. The inverse QFT is a crucial step that enables the efficient extraction of the encoded phase. Mathematically, the state after applying the inverse QFT is described in Equation 29.

$$|\theta\rangle \otimes |\psi\rangle. \tag{29}$$

Finally, the control register is measured, yielding a binary approximation of the phase θ. Figure 13 shows an example of a possible QPE circuit.

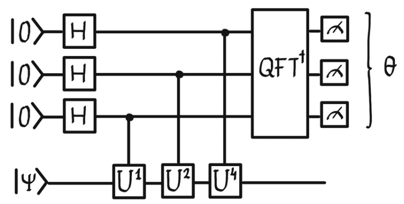

Fig. 13: The quantum phase estimation circuit.

A.2 Types of Probabilities distribution

Several different types of probability distributions can be drawn from the hiphop samples:

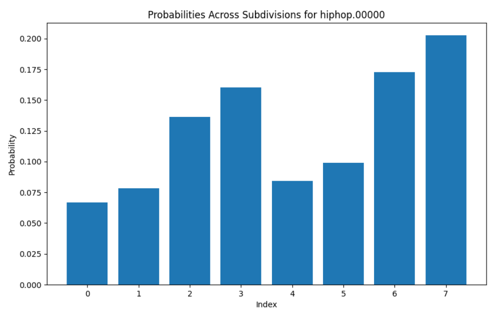

Fig. 14: Probability distribution of sample hiphop.00000

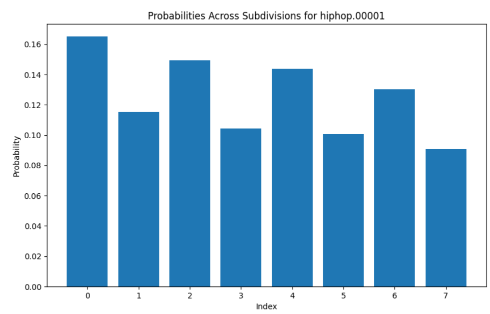

Fig. 15: Probability distribution of sample hiphop.00001

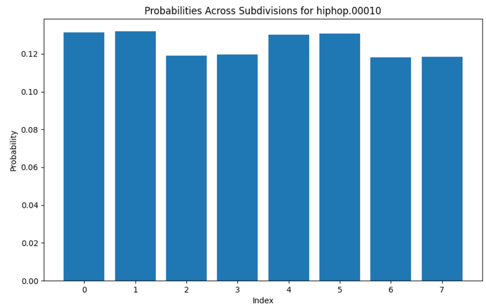

Fig. 16: Probability distribution of sample hiphop.00010

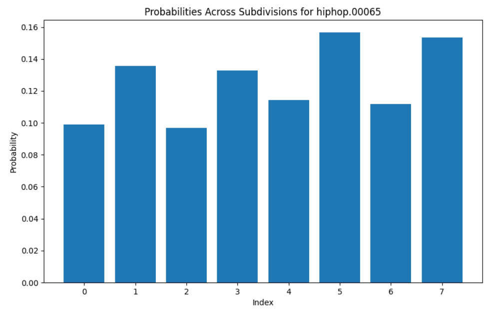

Fig. 17: Probability distribution of sample hiphop.00065

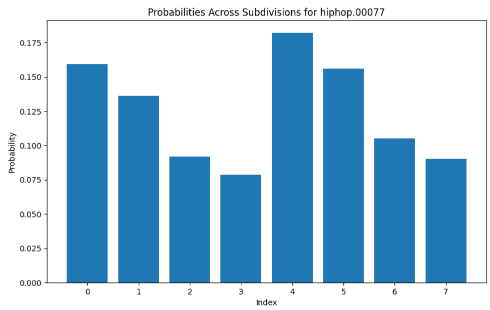

Fig. 18: Probability distribution of sample hiphop.00077