UNIVERSITY OF CALIFORNIA

Los Angeles

AI-Generated Music: Scoring Modern Media

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Music

by

Elliona Ziyan Song Li

2024

ABSTRACT OF THE DISSERTATION

AI-Generated Music: Scoring Modern Media

by

Elliona Ziyan Song Li
Doctor of Philosophy in Music
University of California, Los Angeles, 2024
Professor Ian Krouse, Chair

This dissertation examines the role of artificial intelligence in music composition, focusing on AI-generated music for modern media. It traces the evolution of AI in music, from early computational experiments to contemporary models like Google's Magenta, OpenAI's Jukebox, and Meta's MusicGen. Central to the research is the fine-tuning and evaluation of MusicGen to align with specific stylistic and creative goals.

The study explores the potential of AI to augment artistic processes while addressing ethical concerns such as copyright and authorship. By combining technical analysis with philosophical inquiry, this work positions AI as a tool to enhance creativity, contributing to the ongoing discourse on the intersection of technology and the arts.

The dissertation of Elliona Ziyan Song Li is approved.

Jeffrey Aaron Burke

Steven Franklin Anderson

Jonathan Ryan Beard

Kay Kyurim Rhie

Peter Richard Golub

Ian Krouse, Committee Chair

University of California, Los Angeles

2024

This one is for my parents, my partner, myself, and my years of sweat and tears

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I am wholeheartedly grateful to my mentors, Dr. Ian Krouse and Dr. Peter Golub, for gently guiding me through graduate school, and to my committee members—Dr. Kay Rhie, Prof. Jonathan Beard, Prof. Jeff Burke, and Dr. Steve Anderson — for their generous input, which encouraged me to think critically about this subject from different perspectives. I deeply thank my parents Song Lu and Nan Li, who have loved and supported me throughout my entire academic journey and have always encouraged me to chase my dreams.

My heartfelt gratitude goes to my partner in life, Dr. Hsuan Ming Yu, without whom this dissertation would not exist. He not only held my hand and supported me through countless mental breakdowns along this journey but also did all the heavy lifting in coding and technical aspects, ensuring this project could come to fruition.

I am immensely thankful to my friends who periodically checked in on me to make sure I was alive and hadn't completely lost my mind. Their care and concern reminded me that I was never alone, even during the most challenging moments.

I also want to thank my collaborator, Yolanda Xing, with whom I worked on the VR game *Land of the Forgotten* and who went above and beyond to create an amazing demonstration video for my dissertation defense.

Lastly, I extend my gratitude to my cat, Ares, for being the best boy in the world. His unwavering emotional support is truly priceless.

VITA

Elliona Ziyan Song Li

EDUCATION

UCLA, Los Angeles, CA —Master of Arts, Music composition, 2021

UCLA, Los Angeles, CA —Bachelor of Arts, Music Composition (Cum Laude), 2016

PROFESSIONAL EXPERIENCE

Cartoon Score Composer, Oct 2007 – Feb 2008

Fellow at Sundance Music Lab, June 2022

ACADEMIC EXPERIENCE

Official International Scholar Liaison, Oct 2014 – Jun 2016, UCLA

Assistant Teacher, July 2016, UCLA

Teaching Assistant, Sept 2019 – March 2024, UCLA

PERFORMANCES AND PUBLICATIONS

Guitar and voice piece "Swaying Flower Petals" published by Rising Tide Music Press, 2020

"Swaying Flower Petals" performed by Eliot Fisk and Michelle Rice, April 2022

CHAPTER 1

Introduction

The intersection of artificial intelligence (AI) and music composition has ushered in an era of unprecedented creative possibilities, but it has also raised potential concerns for composers and other creative professionals. As generative artificial intelligence [1] tools become more advanced and accessible, they are being leveraged to generate music, often without the consent or input of the original creators whose works are used to train these models. This practice, commonly referred to as "scraping," poses a serious threat to intellectual property rights, creative autonomy, and the livelihoods of composers. For example, Spotify with its infamous "fake" artist scandal in the early 2020s (10). Generative AI is rapidly being adopted across a wide range of industries, including those that employ creative workers, posing a potential for AI to displace jobs (11). In this landscape, it is crucial to empower individual creative workers with tools that enhance their artistry and safeguard their agency and intellectual ownership. This technology is still in its early stages, with ongoing efforts to establish ethical and legal frameworks. As these aspects continue to take shape, it remains to be seen which challenges will emerge and how they will be addressed over time.

Central to this study is the premise that AI technologies, if designed and implemented thoughtfully, could democratize access to high-quality compositional resources, empowering independent creators who may lack the extensive infrastructure and funding available to

[1] A branch of artificial intelligence that creates new content, such as text, images, music, or code, by learning patterns and structures from existing data and generating outputs that mimic those patterns

larger organizations. However, this optimism is tempered by the reality of systemic barriers. Big Tech companies and established composer collectives often dominate the technological and creative landscapes, restricting access to advanced tools and high-profile opportunities.

This dissertation is driven by the goal of finding an open-source AI music generation model that can be run locally and fine-tuned to emulate the stylistic nuances of specific composers. Such a model would not only democratize access to AI tools but also allow composers to retain control over their creative processes and outputs. By allowing individual composers to customize and adapt these technologies to their unique artistic visions, we can reclaim the creative agency that is increasingly ceded to large technology corporations.

The focus on local implementation is intentional and strategic. Local models eliminate reliance on cloud-based systems, often proprietary and opaque about their training data and processes. This lack of transparency from some of the commercialized models has fueled concerns about unethical scraping practices and the exploitation of creative works. On the other hand, a locally run, fine-tuned model ensures that composers can work in a self-contained environment where their data and intellectual properties remain protected. Through this lens, the dissertation addresses both the promises and limitations of current AI technologies, acknowledging their reliance on training data and the absence of the intuitive creativity that defines human artistry.

Furthermore, such a model would be accessible to everyday composers, not just those with extensive technical expertise or substantial resources. This accessibility is critical in leveling the playing field, enabling independent creators to leverage the same technological advancements reshaping the music industry. This work also examines how AI can serve as a collaborative partner, capable of augmenting creative processes rather than replacing them. For instance, AI systems can handle repetitive or labor-intensive tasks, enabling composers to focus on higher-level artistic decisions. By integrating such tools into workflows, independent composers may gain access to resources traditionally reserved for large-scale productions, creating opportunities for experimentation and innovation. By fostering a more equitable landscape, composers can adapt to the challenges of the AI age without losing their distinct voices or being overshadowed by corporate interests.

The creative industries are at a pivotal juncture where the rapid evolution of AI demands a collective response to preserve artistic integrity. This dissertation represents one step in that direction: exploring, testing, and refining an AI model that prioritizes accessibility, adaptability, and ethical considerations. It is an attempt to not only embrace the potential of AI as a collaborator in the compositional process but also to protect and empower the human creators who drive the art form forward. In doing so, we ensure that technology tools serve the artist, not the other way around.

This work is intended as the first part of a larger project. In the future, this research will aim to develop an AI model capable of generating real-time musical responses tailored to what the experiencer sees on screen in an immersive environment. Such a model would push the boundaries of interactivity and adaptability in music, enhancing the depth and personalization of immersive experiences in virtual and augmented realities. By laying this groundwork, this dissertation sets the stage for future advancements in AI-driven music composition.

CHAPTER 2

The start of (almost) everything

2.1 The pursuit of total immersion

Since the advent of the technology that enabled its feasibility, the human aspiration to transcend the mundanity of daily existence has manifested in a constant pursuit of immersion within fantastical realms. This desire for escapism is apparent not just in the building of physical spaces such as amusement parks, intended to give a temporary reprieve from the quotidian through meticulously crafted, immersive experiences, but also in the engagement with virtual spaces, such as those produced by Virtual Reality (VR)[1]games. It might be posited that the action of conjuring narratives in one's imagination represents the primordial iteration of virtual world-building, showcasing an intrinsic human propensity towards creative escapism.

Dreams represent a complicated form of world-building that is autonomously executed by our brains, a method that unfolds without conscious effort. Psychological theories, such as those suggested by Freud, claim that dreams enable people to fulfill unmet desires and symbolically process unresolved conflicts. This unconscious processing can offer emotional relief and add to psychological well-being, providing an escape from everyday stresses and emotional burdens. It is thought that dreams function as a mirror, reflecting the complex

[1] An immersive technology that simulates a three-dimensional, interactive environment, allowing users to engage with a computer-generated world through specialized hardware like headsets and controllers

tapestry of our psychological states and as a window, revealing probably the most profound aspirations harbored within us (12). Moreover, dreams are a critical element of the day's processing of information(13) Cognitive theories of dreaming propose that dreams provide an adaptive function by allowing the rehearsal of threat perception and problem-solving strategies, effectively allowing the person to learn coping mechanisms and check out solutions with no real-world consequences.

As an alternative to dreams, or perhaps even an attempt to dream "on demand", many people use recreational drugs as a way to temporarily get away from the pressures and also monotony of their daily lives. Recreational drug use usually enhances sensory perceptions, resulting in novel experiences that starkly contrast with reality. The pharmacological effects of psychedelics (e.g., LSD, psilocybin) are identified to alter perception, thought, and emotion, which could make the usual appear extraordinary. This pursuit of novelty may be known as an endeavor to enjoy realms of experience beyond the access of typical cognitive functions. In specific cultural contexts, the usage of particular drugs for leisure purposes is embedded in interpersonal rituals offering a communal kind of escapism. The application of medications for leisure uses by people dates to ancient times, with evidence suggesting that such methods have been a component of human activities thousands of years ago. Archaeological findings and historical records indicate that early civilizations, like those in China, India, Egypt, and Mesopotamia, used different organic materials, such as opium, cannabis, and coca leaves for religious, medicinal, and recreational purposes. For instance, evidence of coca leaf chewing, which includes psychoactive alkaloids, grounds for cocaine, goes back more than 5000 years in South America (14). Similarly, cannabis continued to be used in rituals and also for leisure for a considerable number of years, with proof of its use dating to, at a minimum, 2500 BC inside the Eurasian Steppe. Opium, derived from poppy seeds, has likewise been utilized since ancient times, with Sumerian texts from around 4000 BC describing it as the "joy plant."

The overarching goal of these endeavors is consistently targeted at facilitating short-term transportation of the person into alternative dimensions, therefore offering a respite from their ordinary lives. This pursuit reflects a deep-seated yearning inside the human psyche for encounters that transcend the limits of the concrete world, underscoring the benefits of virtual and imaginative constructs in enriching human knowledge.

In discussions of virtual world-building, there is a primary focus on the visible features of building electronic environments, often at the cost of the aural dimension. This oversight is essential despite the historic acknowledgment of the benefits of combining several sensory experiences in art technique, as exemplified by Richard Wagner's idea of Gesamtkunstwerk, released in 1849. Wagner's idea advocated for a "total work of art" that synthesizes visual elements, theater, and music, underscoring the prospective depth and richness that aural elements can contribute to an immersive experience.

With the arrival and development of technical innovations, "immersive media" has frequently become associated with visual-centric products, such as VR headsets. These devices, emblematic of the electronic era, provide users instant access to virtual realms without joining physical venues such as theaters. This shift towards readily accessible, visually immersive experiences reflects a broader pattern in media usage, in which convenience and immediacy usually take precedence.

The genesis of such immersive media extends further back in history than is frequently recognized, with its precursors identifiable within the visionary narratives of early 20th-century science fiction literature. For instance, Laurence Manning's seminal 1933 narrative, "The Men Who Awoke," envisages a unit replicating human sensory experiences through electric stimulation, foreshadowing contemporary VR experiences.

This literary foresight is the forerunner for the following empirical developments in im-

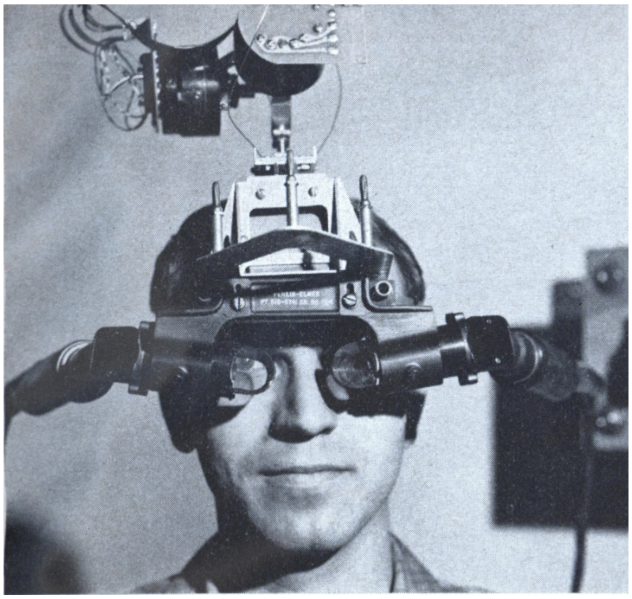


Figure 2.1: The Sword of Damocles (7)

mersive media. A pivotal milestone was accomplished with Morton Heilig's development of the Sensorama in the 1950s, an early attempt at multi-sensory integration in media. Additionally, Ivan Sutherland's conceptualization of the "Ultimate Display" – *The Sword of Damocles* in the 1960s (15) marked a significant leap towards interactive and immersive digital environments, setting the stage for future innovations in VR.

The video game sector has additionally played an essential part in the evolution of immersive technologies. Notable examples are the conception of Sega VR Powered Shades and the release of Nintendo Virtual Boy during the 1980s and 1990s. Although the former was never released and the latter was not a commercial success, they symbolize people's pursuit of total immersion, regardless of how ineffective the end products may be. These developments underscore the video game industry's contribution to the broader trajectory of immersive media, illustrating a continuum of features that bridges early speculative fiction and contemporary electronic realities. This historical arc highlights the symbiotic relationship between imaginative literature and technological advancement, underscoring the profound effect of visionary narratives on the materialization of virtual environments.

Nevertheless, this particular emphasis on visual technology potentially neglects the multifaceted design of immersive experiences, where sound plays a crucial part in engendering a fully realized, immersive setting. Sound not only enhances the realism and depth of virtual worlds but also evokes emotional responses and aids in narrative storytelling within these digital landscapes. The integration of high fidelity, spatial audio with visual pieces in virtual spaces could appreciably boost the general feeling of presence and immersion, a principle which harks to Wagner's holistic approach to art with music being an additional dimension in this attempt to totally immerse the audience in the art.

In the domain of visual, or more accurately described, multi-sensory media, music often remains an underestimated component. Music not only plays a pivotal role in narrative storytelling but also possesses the capacity to subtly influence the audience's emotional state, directing their feelings at precise moments. Appropriately utilized, music transcends its ancillary status, acting as a potent enhancer of the media experience. Some advocates contend that music introduces an additional dimension to a creative work, whether in cinema or video gaming. Applications and services such as SonicMaps have begun to leverage "background music" to enrich users' daily routines. This apprehension towards music's potent influence is precisely why some filmmakers approach its incorporation with caution, always mindful of its potential to overshadow or unduly sway the narrative. This cautiousness underscores the broader discourse on music's place in immersive media environments. When contemplating fully immersive media experiences, the prospect of crafting real-time, custom scores that adapt to each user's decisions within a virtual environment presents an intriguing avenue for creating uniquely personalized experiences. Achieving such a feat necessitates leveraging artificial intelligence to generate a copious repertoire of music, circumventing the constraints posed by the slow pace at which humans compose – essential for real-time musical adaptation—and endurance, given the exhaustive nature of continuously creating new compositions for identical scenes. Fortuitously, the intersection of composition and science has witnessed a burgeoning interest in the exploration of AI-generated music, and the exploration of AI assisted musical composition has been underway for some time, heralding a new frontier in dynamic and adaptive musical scoring.

2.2 Meanwhile in AI music

Since the 1950s, composers and computer researchers have begun exploring the production of music using computer programs. Among the seminal moments within the confluence of technology and music, came the composition by a computer – *The Illiac Suite* for string quartet in 1957. This pioneering work was facilitated by the ILLIAC I computer at the Faculty of Illinois, marking a significant event in using computational assets in music. As the consequence of an interdisciplinary collaboration between Lejaren Hiller (a composer/chemist) and Leonard Isaacson (a mathematician,) this project stands as a landmark in the history of digital music, frequently cited as the inaugural instance of a computer being utilized to compose music in a substantive manner.

The piece has four movements, each one the result of an experiment. To start their experiments, Hiller and Isaacson tasked the computer to write basic melodies. They employed a method that allowed the machine to produce random numbers, utilizing a strategy inspired by the "Monte Carlo" method - an approach originally developed by physicists to address complex problems characterized by numerous probabilities. Random integers ranging from 0 to 14 were assigned to correspond with the pitches[2] across a two-octave span of the C-major scale, focusing solely on the white notes. They then "selected the rules from the elaborate

[2] Excluding sharps and flats

injunctions for "strict first-species counterpoint" (8). In the preliminary stage of the algorithmic composition process, integers that effectively met the critical elements established by the note screening protocol had been initially retained within the computer's mind. This retention policy dictated that such integers have been withheld from external output until the computational apparatus had concluded a coherent melody characterized by its initiation and conclusion on C. Regularly, the personal computer experienced issues in determining a note that conformed to the established permissive criteria. Upon encountering 50 such problems, the method was programmed to expunge the nascent melody from its memory, resetting the compositional process. It could produce many simple melodies ranging from three to twelve notes within an hour. More instructions have been added to enable the laptop to create two voice counterpoints while screening out the dissonances between notes, and therefore, the very first movement came to be.

In the second movement, Hiller and Isaacson enhanced the algorithmic framework by incorporating an augmented suite of screening protocols that encapsulated fourteen rules derived from the first species counterpoint in four voices. In this evolved computational experiment, the mechanism continued to generate random pitches, specifically limiting its selection to the diatonic tones analogous to the white notes employed in the initial study. However, in this iteration, the inherent randomness was algorithmically constrained to engender a degree of redundancy. Consequently, the emergent melodies, despite being constrained to whole notes only, in the style of Palestrina.

In the third movement (and also the third experiment,) Hiller and Isaacson increased the rhythmic and dynamic complexity. A straightforward approach yielded a significant diversity in rhythm for them. By adopting 4/8 time as the meter and designating the eighth note as the smallest rhythmic unit, they encoded all possible rhythmic patterns within these constraints using binary digits. For instance, the sequence 1111 signified four eighth notes;

1110 indicated two eighth notes followed by a quarter-note; 1010 denoted two quarter-notes, among other permutations. These permutations generated a series of binary numbers corresponding to decimal numbers ranging from 0 to 15. Given that rhythmic changes do not typically occur at every measure in music, they introduced rhythmic redundancy through a secondary series of random numbers, programming the computer to repeat a specific rhythm for up to 12 iterations. Alongside this "horizontal" redundancy[3]within the melodic lines of individual voices, an additional binary code [4] implemented "vertical" (homophony[5]) redundancy across the four voices, whereas the code 0000 meant that all four voices would operate rhythmically independent of one another, while 1111 mandated homo-rhythm across all voices, among other variations. They also applied similar methodologies to incorporate patterns of dynamics and variations in playing instructions(8). The resulting work sounds drastically different from the first two movements, especially when the computer was permitted to generate random chromatic notes,[6] it was producing music that was highly dissonant[7]. The resulting composition sounds unpleasant due to the lack of artistic logic behind note selections.

In the last movement, they sought to write a movement that was purely based on mathematical rules. As Hiller describes, "... the computer was programmed to select the intervals between successive notes according to a table of probabilities instead of at random . Moreover, the probabilities themselves were made to shift in accordance with so called Markov probability chains."(8) In this context, a Markov chain is a mathematical system that undergoes transitions from one state to another on a state space. It is "memoryless," meaning

[3] Repetition, Hiller preferred the term "redundancy"

[4] Limited the rhythm to homophonic, chords style, hymn style, or homo-rhythmic

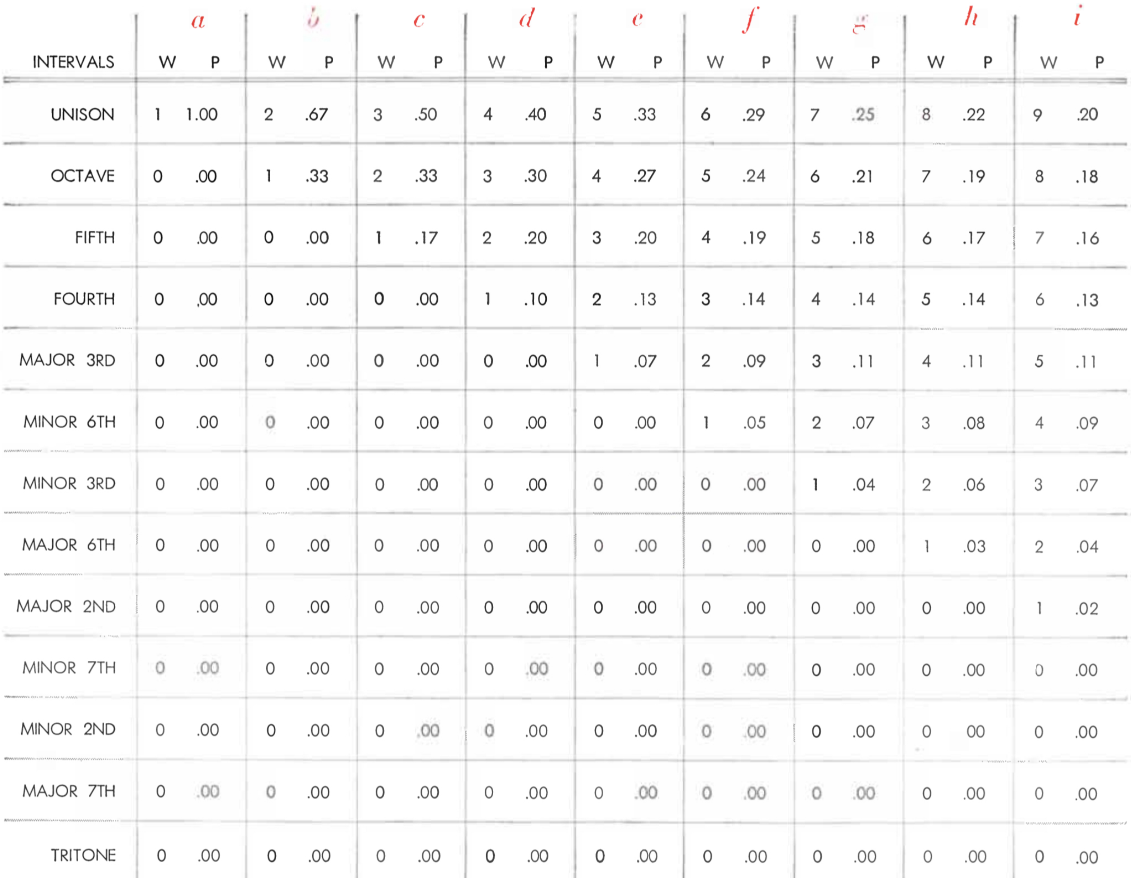
[5] A musical texture where multiple voices or parts move together rhythmically

[6] Pitches that lie outside the diatonic scale of a given key

[7] When a combination of notes or chords create a sense of tension, instability, or harshness

the next state depends only on the current state and not on the sequence of events that preceded it. Here, each "state" represents a specific musical interval between notes. For example, in the initial phase of the composition, the unison interval (where two successive notes are the same, creating a "zero" interval) is assigned a weight of one, indicating it has a probability of occurring. In contrast, all other intervals start with a weight of zero, meaning they initially have no chance of being selected. As a result, all voices in the composition stay on the same note. After two bars, the composition's rules change to increase the unison's weight to two and introduce the octave interval with a weight of one. This adjustment makes the unison interval twice as likely to occur as the octave. As the composition progresses, additional intervals are introduced with their weights adjusted to change their probabilities of occurring. For instance, the fifth interval is added next, and the weights of the unison, octave, and fifth are adjusted to three, two, and one, respectively, reflecting their likelihood of occurrence. The process continues with the introduction of new intervals every two bars, and the weights are reassigned to reflect the changing probabilities of each interval being chosen. This method allows for a structured yet probabilistic approach to musical composition, where the sequence of intervals evolves in a controlled manner, influenced by the predetermined weights and adjustments over time.

After the Illiac Suite's "premiere" on Aug. 9th, 1956, Hiller became famous overnight. Some people did not like the beat, and some performers thought the piece (especially the 4th movement) could be unnatural at times, and "had a quirkiness that throws you off."(16) The performance was met with polarized reactions, centering around the controversies of the broader implications of computer-generated music. The idea that a computer could compose music challenged traditional notions of creativity, authorship, and the role of the human composer. Critics questioned whether music generated by algorithms could possess the emotional depth and artistic value of music composed by humans. There was skepti-



	a		*b*		*c*		*d*		*e*		*f*		*g*		*h*		*i*	
INTERVALS	W	P	W	P	W	P	W	P	W	P	W	P	W	P	W	P	W	P
UNISON	1	1.00	2	.67	3	.50	4	.40	5	.33	6	.29	7	.25	8	.22	9	.20
OCTAVE	0	.00	1	.33	2	.33	3	.30	4	.27	5	.24	6	.21	7	.19	8	.18
FIFTH	0	.00	0	.00	1	.17	2	.20	3	.20	4	.19	5	.18	6	.17	7	.16
FOURTH	0	,00	0	.00	0	.00	1	.10	2	.13	3	.14	4	.14	5	.14	6	.13
MAJOR 3RD	0	.00	0	.00	0	.00	0	.00	1	.07	2	.09	3	.11	4	.11	5	.11
MINOR 6TH	0	.00	0	.00	0	.00	0	.00	0	.00	1	.05	2	.07	3	.08	4	.09
MINOR 3RD	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	1	.04	2	.06	3	.07
MAJOR 6TH	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	1	.03	2	.04
MAJOR 2ND	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	1	.02
MINOR 7TH	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
MINOR 2ND	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	00	0	.00
MAJOR 7TH	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
TRITONE	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00

Figure 2.2: Markov Probability table for Intervals(8)

cism about the authenticity and artistic value of computer-generated music. Some argued that music, as an expression of human emotion and experience, could not be authentically replicated by machines. For some musicians and composers, the advent of computer-assisted composition raised fears about the obsolescence of human composers. There was concern that machines might replace humans in the creative process, a theme that has recurred with advances in AI and automation across various fields. The "Illiac Suite" and subsequent computer-generated compositions often sparked debate over the balance between technical innovation and artistic merit. While the technical achievements were acknowledged, some critics were not convinced that the resulting compositions met the standards of high quality art music. To me, this piece, though it may be mathematically sound, resembles many compositions by human composers from an era when many people approached music theory as if it were a STEM subject. It does not sound musical to me and feels as though it lacks the depth and soul of human expression.

Despite these controversies, the "Illiac Suite," as well as the work of Isaacson and Hiller, have demonstrated the possibility of computer systems in creative processes. They opened up new avenues for exploration in algorithmic composition, resulting in additional advancements in computer music and the broader area of electronic arts. Over time, as technology advanced and society's connection with computer systems deepened, the acceptance of computer-generated music and its potential for creative works has continued to grow, though discussions about the role of imagination and technology in the arts persist.

2.3 Fast forward a few decades

The development of AI in music composition evolved through various phases, incorporating different technologies and methodologies. In the 1970s and 80s, researchers experimented with rule-based systems and early forms of machine learning to create compositions and to assist in music production. These systems often relied on predefined rules or algorithms to generate music based on specific styles or parameters.

AI differs from traditional algorithmic composition primarily in its approach to generating music. Algorithmic composition relies on predefined rules, parameters, or mathematical formulas explicitly programmed by the creator. These systems can produce music that follows specific styles or structures but are limited by the rigidity of their design. AI, particularly machine learning models, goes beyond this by analyzing large datasets of music to learn patterns and structures autonomously. This ability allows AI to generate outputs that reflect a deeper understanding of style and context, often with results that can appear more organic or innovative. For example, while a rule-based system might generate a melody by following a fixed set of harmonic rules, an AI model could synthesize new material by drawing on patterns it learned across thousands of compositions, adapting its output to nuances it has inferred but not explicitly been taught.

The advent of Musical Instrument Digital Interface (MIDI[8]) in the early 1980s was a significant milestone, as it facilitated easier communication between electronic musical instruments and computers, enabling more sophisticated compositions and experiments with AI-generated music.[9]

"Artificial Intelligence and Music" by Curtis Roads, published in 1980, marks a critical moment in exploring the intersection between AI and music (17). It presents AI as a new paradigm capable of transforming various aspects of music, including composition, analysis, performance, and even the cognitive processes involved in musical tasks. Roads discusses how AI introduces methodologies that offer new strategies for addressing musical problems and provide deeper insights into cognitive processes without necessarily trying to replicate human mental activity.

The paper outlines the history of automated music-making, beginning with early mechanical instruments, such as carillons and automata (18) and moves through more recent innovations like algorithmic composition and early computer music experiments. Roads emphasizes that AI in music is not an entirely new concept; centuries-old examples of rule-

[8] A standardized protocol that allows electronic musical instruments, computers, and other devices to communicate, control, and exchange musical information such as notes, dynamics, and tempo

[9] The generation of sound files and symbolic compositions serves different purposes in the creative process, and each has unique implications. Symbolic compositions, such as MIDI files or sheet music, represent the underlying structure of music: the notes, rhythms, and dynamics, detached from specific timbral qualities or production choices. These are akin to blueprints that can be interpreted by performers or further processed by software to render the final audio. Generating sound files, on the other hand, involves producing fully rendered audio, complete with instrument timbres, spatial qualities, and effects. This is a more immediate output, often bypassing the need for interpretation or additional production.

based generative music, such as Guido d'Arezzo's pitch generation method[10] and Mozart's dice game, [11]demonstrate a long-standing interest in formalizing and automating musical processes, and can be considered as forerunners of generative music(17).

Roads divides his discussion into two main strands: efforts toward musical intelligence and applications of AI methodology. He reviews several early projects that combined AI with music, such as Lejaren Hiller's algorithmic compositions, which treated music as an algorithmic process and initiated the use of computers for composition. Early AI applications in music included the development of systems for music analysis, such as Simon and Sumner's 1968 work on pattern recognition in music and Winograd's harmony analysis program, which applied systemic grammar to analyze tonal harmony (19).

The paper also explores how generative modeling, a branch of AI that seeks to model existing musical structures rather than create new compositions, became essential for understanding the rules governing music. These AI models helped theorists test hypotheses about the underlying structures of specific musical traditions, such as tonal music and provided a new way to compare different musical styles.

Throughout the paper, Roads examines the limitations of early music analysis programs, which often only focused on surface-level features of compositions. He suggests that AI could address these limitations by incorporating a more profound knowledge of the music being analyzed, potentially leading to a more comprehensive and insightful analysis. The paper also introduces more sophisticated AI-driven systems that aim to recognize, understand, and generate music, drawing from cognitive theories and AI concepts like rule-based systems and grammar models.

[10] Guido d'Arezzo created a table-lookup method for producing pitches from spoken words, which is an early illustration of a generative procedure for composition(17)

[11] Mozart's *Musikalisches Würfelspiel*, which is a system for generating music using dice, is one example of an early generative technique for composition(17)

In terms of future directions, Roads envisions that AI will play a role in developing intelligent music systems capable of analyzing, composing, and performing music in ways that transcend the capabilities of human musicians. These systems, he argues, could handle tasks such as intelligent sound analysis, the creation of dynamic musical performances, and the development of new musical machines. He also touches on the societal implications of AI in music, questioning whether such technologies will enrich musical creativity or merely substitute mechanical performances for human artistry.

2.4 Fast forward a few more decades

In the 1990s and 2000s, more advanced AI techniques were introduced, including neural networks[12] and evolutionary computing,[13] allowing for more complex and expressive musical creations. David Cope's "Experiments in Musical Intelligence" (EMI, also known as "Emmy") is a significant and pioneering project in the field of AI-generated music (20). Developed by Cope, a composer and professor at the University of California, Santa Cruz, EMI was designed to analyze the music of various classical composers and then generate new compositions in the style of those composers. Cope began working on EMI in the 1980s during a personal creative block while writing an opera. His initial goal was to create a tool that could assist him in analyzing his style in hopes of finding his undiscovered music. However, the project quickly evolved into an exploration of the possibilities of algorithmic composition. EMI was built using various techniques, including pattern matching, rule-based systems, and, later, more sophisticated AI methodologies.

[12] Computational models inspired by the structure and function of the human brain, consisting of interconnected layers of nodes (neurons) that process and learn from data to recognize patterns, make predictions, or perform tasks

[13] A branch of artificial intelligence that uses algorithms inspired by biological evolution, such as natural selection and genetic variation, to solve optimization and complex computational problems

Pattern matching in AI involves algorithms identifying and using recurring patterns within data to make decisions, predictions, or generate new data. It is fundamental in various AI applications, from natural language processing, where it helps understand and generate text, to AI-generated music, and central to many systems that analyze, compose, or improvise music. It involves the algorithm identifying, analyzing, and utilizing patterns within musical compositions to generate new pieces of music that maintain stylistic consistency with the source material. This process is pivotal in various applications, from generating new compositions in the style of specific composers to improvising jazz solos.

In music, a "pattern" can refer to a wide range of musical elements, such as a sequence of notes (melody,) rhythms, harmonies, or even the structural aspects of compositions (like the form of a piece.) Pattern matching involves the algorithm recognizing these elements in existing music and understanding their relationships and contexts.

The first step involves analyzing a corpus of music to identify recurring patterns. This can be a database of compositions by a specific composer or within a particular genre. The AI system breaks down the music into manageable components, such as phrases, chords, or motifs, and looks for recurring structures or sequences. Once patterns are identified, the system classifies them based on various musical features. This might involve categorizing motifs by their rhythmic qualities, harmonic functions, or melodic contours. The recognized patterns are stored in a database, with metadata describing their musical characteristics and contexts. This database is the foundation for the system's musical "vocabulary." The system uses the stored patterns as building blocks when generating new music. The recombining, modifying, or extending these patterns can create new musical pieces that are stylistically coherent with the source material. Many systems incorporate a feedback loop where generated music is evaluated against certain criteria (like stylistic consistency, coherence, and novelty) and adjusted accordingly. This might involve refining pattern selection or the way patterns are combined.

Machine Learning Algorithms like Hidden Markov Models (HMMs) offer a robust framework for generating music by learning the underlying statistical structure of musical compositions, such as musical motifs, phrases, or chord progressions, and the model understands how the transitions between states capture the musical flow or changes in motifs[14]/phrases. Given a trained HMM, generating new music involves starting from an initial state and then moving through states based on the transition probabilities, emitting notes or chords based on the emission probabilities. This can create music that follows learned patterns and structures. By training an HMM on a specific collection of music, the model can learn to generate music that mimics that style. The model captures the underlying statistical properties of the music, such as typical note sequences, rhythm patterns, and harmonies. One of the limitations of HMMs is the assumption of the Markov property that the future state depends only on the current state and not on the sequence of events that preceded it. This can be overly simplistic for some musical structures that need to depend on more prior states. Neural networks and decision trees[15] can be used to identify and classify patterns. Some systems rely on predefined rules grounded in music theory to identify and recreate patterns. Genetic Algorithms: these can evolve musical patterns over successive generations, selecting and recombining "most likely" patterns to create new compositions.

While pattern matching is a powerful tool in AI music generation, it also presents challenges, such as ensuring that generated music is a mere imitation of recognizable patterns and contains originality and creativity. Balancing coherence with novelty and managing the vast diversity of musical expression are ongoing areas of research and development in the

[14] A short, recurring melodic, rhythmic, or harmonic idea that serves as a foundational element in a composition

[15] A type of machine learning model that use a tree-like structure of decisions and their possible outcomes to classify data, make predictions, or solve problems by breaking them down into smaller, simpler decisions

field.

Genetic algorithms provide a biologically inspired approach to algorithmic music composition, utilizing principles like selection, crossover, and mutation to generate and refine musical material. In this method, musical elements such as notes, rhythms, and harmonies are encoded into data structures that act as musical "chromosomes" or "genomes." An initial population of these chromosomes, often created randomly, represents a variety of potential musical ideas. Each chromosome is evaluated using a predefined fitness function, which assesses how well it meets specific musical criteria or stylistic goals, such as adherence to a particular scale or rhythm. Chromosomes with higher fitness scores are more likely to be selected for reproduction, contributing their traits to the next generation. Through crossover, these selected chromosomes combine their musical data to produce offspring that inherit traits from both parents, while random mutations introduce additional variation, ensuring diversity within the population and avoiding stagnation. This process of selection, crossover, and mutation is repeated over multiple generations, gradually evolving a population of musical ideas that better meet the desired criteria. The algorithm ultimately outputs the best-performing chromosome or a selection of high-fitness chromosomes as the final musical composition or a set of promising ideas. EMI begins by analyzing the input compositions, breaking them into smaller components to identify recurring patterns and structures. It then recombines these elements in new ways, ensuring the newly generated music adheres to the stylistic rules and constraints extracted during the analysis. Finally, the generated compositions are evaluated based on how well they follow the style and maintain coherence. The system may iterate this process several times, refining the output until it meets predefined criteria(21). While genetic algorithms are a powerful and flexible tool for generating diverse musical outputs, they require careful design of fitness functions and algorithm parameters, necessitating both musical insight and computational expertise .

EMI generated considerable interest and debate within both the music and AI communities. Some of Cope's compositions created by EMI were performed in concert halls and even recorded, with audiences often unable to distinguish between the AI-generated pieces and genuine compositions of historical composers. However, EMI also sparked controversy and philosophical debates about creativity, originality, and the role of AI in art. Critics questioned whether music generated by an algorithm could possess the emotional depth and expressiveness of human-composed music. Others saw EMI's achievements as a demonstration of the potential for AI to extend and enhance human creativity.

CHAPTER 3

Modern attempts

The 2010s and 2020s witnessed a surge in the capabilities and applications of AI in music, driven by advancements in machine learning, particularly deep learning.[1] Projects like Google's Magenta, OpenAI's Jukebox, MuseGAN, and IBM's Watson Beat showcased the potential of neural networks to generate music that could capture the nuances of human composition and even mimic specific genres and artists' styles. This chapter serves as the traditional "literature review" section, providing an overview of the technologies and models discussed in existing research. It is important to note that the models and methodologies presented here are based on secondary sources and have not been personally tested or implemented as part of this study.

3.1 Neural networks and decision trees

One paper explores the different possibilities when it comes to generating music using neural networks. "Experiments in Modular Design for the Creative Composition of Live Algorithms" by Oliver Bown explains the concept of live algorithms (LAMs) in music composition, particularly focusing on their modular design to enable real-time interaction between musicians and algorithmic systems (22). The idea behind live algorithms is to create compu-

[1] A subset of machine learning that uses artificial neural networks with multiple layers to automatically learn and extract complex patterns and representations from large amounts of data

tational systems that exhibit a degree of autonomy in music performance, either alongside human musicians or on their own. These systems are not meant to mimic human musicianship but rather introduce novel, interactive, and autonomous behaviors into musical performance.

The paper discusses a modular design framework proposed by Blackwell and Young, where live algorithm systems are broken down into three key components: analysis (P), synthesis (Q), and patterning processes (I). These components allow for the possibility of substituting one module for another, creating flexible and interchangeable parts within the system. The goal is to facilitate a creative, dynamic, and collaborative approach to algorithmic music composition.

Two case studies are presented in the paper, each focusing on the use of different algorithms for driving real-time musical improvisation systems. The first study uses Continuous-Time Recurrent Neural Networks (CTRNNs) while the second employs Decision Trees (DTs.) Both systems aim to create a level of creative autonomy by producing generative musical behaviors that respond dynamically to incoming musical data. CTRNN is a neural network model, a form of Recurrent Neural Network (RNN,) which is a type of artificial intelligence model designed to handle data that comes in sequences, like a sentence, a song, or a time series. In simple terms, an RNN is like a smart memory system that processes sequences one step at a time, using what it has already learned to make better predictions about the future. What makes RNNs special is that they can remember information about what happened earlier in the sequence and use that memory to help make decisions about what comes next.[2] Imagine you're reading a sentence, and each word helps you understand the next one. RNNs work similarly—they process one piece of data at a time, and each step builds on what

[2] As opposed to with HMMs, where the next state depends only on the current state and not what proceeds it

the network has already seen. So, if you're trying to predict the next word in a sentence, the RNN looks at the previous words and uses that info to guess what's coming next.

The "recurrent" part means the network loops back on itself. This looping lets the network remember past information for a while. However, one limitation of basic RNNs is that they can struggle to remember things from far back in a long sequence. That's why more advanced versions, like Long Short-Term Memory (LSTM,[3]) or Gated Recurrent Units (GRUs,) were created to improve how they handle longer-term memory.

In generative AI, GRUs are widely used due to their ability to efficiently model temporal dependencies, which are crucial in the sequential nature of music. A GRU is a variant of a RNN architecture designed to manage long-term dependencies in sequence data while avoiding the vanishing gradient problem,[4] which typically hampers the performance of traditional RNNs. Unlike standard RNNs, GRUs use gating mechanisms to regulate the flow of information, allowing the model to selectively retain important musical patterns and discard irrelevant data. This functionality makes GRUs particularly useful in music generation tasks, where understanding the temporal structure and maintaining coherence over time is essential.

In AI music generation, various models harness the power of GRUs for creating musical sequences. While Generative Adversarial Networks (GANs,[5]) as seen in models like MuseGAN, focus on generating polyphonic music across multiple tracks, GRUs have been employed in the generation of monophonic melodies and chord progressions due to their efficiency and simplicity. For instance, models such as DeepBach, which is designed to generate

[3] Which we'll get to when we talk about

[4] A problem in deep learning where the updates to the earlier layers of a neural network become too small, making it hard for the network to learn properly

[5] Two networks: a generator and a discriminator, which compete with each other in a game-like setting

harmonization in the style of Johann Sebastian Bach, or Google's MusicVAE, which works with latent variable modeling for music sequences, can benefit from the temporal handling of GRUs, even if GRUs are not the primary architecture in these systems.

A GRU consists of two primary gates: an update gate and a reset gate. These gates help the model decide when to update the hidden state with new information and when to reset it, allowing the GRU to maintain relevant musical context over long sequences. This is particularly valuable in tasks like melody continuation, where the model needs to predict the next note in a sequence based on the preceding notes, preserving the overall structure of the music while generating something that sounds musically coherent and pleasing. GRUs are thus an essential tool in music generation models that require a balance between learning long-term dependencies and computational efficiency. This is crucial for generating high-quality, coherent music that maintains both rhythmic and harmonic consistency over time.

Back to CTRNN (page 23,) the CTRNN which in Bown's paper is designed to generate complex, reactive behaviors through a network of interconnected artificial neurons. Each neuron in a CTRNN processes and transmits floating-point values through weighted synapses, which connect it to other neurons. These neurons continuously update their states by summing the weighted inputs they receive and producing an output that is then passed along to other neurons. This process happens quickly and synchronously, creating a smooth, continuous flow of activation throughout the network.

The outputs of CTRNN neurons are typically constrained to values between -1 and 1 by applying a sigmoid function to the output of each neuron. This results in what can be seen as a "black box" system: a CTRNN takes a set of real-valued inputs, processes them, and produces a set of real-valued outputs, all within the defined range. CTRNNs have been used successfully in musical contexts to create dynamic, compelling patterns, especially for tasks that involve continuous modification of sound parameters, such as altering playback position in granular synthesis or controlling parameters in FM synthesis. However, due to their complexity and continuous behavior, CTRNNs are not well-suited for music tasks that require precise solutions, like harmonization or structured melodic patterning. They are more appropriate for creative tasks that benefit from fluid, evolving behavior.

The first study with CTRNNs focuses on the ability of these networks to generate continuous,[6] reactive behaviors in response to external stimuli. By evolving CTRNNs through a genetic algorithm,[7] the study seeks to create networks that can activate and rest dynamically, responding to changes in the musical environment. This behavior is akin to a "dynamical reservoir," where the system remains active while external stimuli are present and eventually returns to rest in the absence of stimuli (22).

The second study explores the use of DTs (page 23,) which are more discrete and interpretable than CTRNNs. DTs are often used for classification tasks and are well-suited for generating distinct musical events based on real-time audio analysis. Unlike CTRNNs, which produce continuous outputs, DTs operate by making decisions at each time step, flipping between discrete states[8]. This allows for more predictable and stable rhythmic patterning, making them more efficient and easier to analyze than CTRNNs. Additionally, the modular structure of DTs allows for more straightforward growth and evolution, making them highly adaptable in live performance contexts.

DTs, like CTRNNs, process real-time audio features, updating every 10 milliseconds. To mimic the continuous behavior of CTRNNs, the DTs include an internal state represented by floating-point values that are analyzed and modified based on decisions made by the tree.

[6] In simple terms, not whole numbers

[7] A problem-solving method that mimics natural selection, where possible solutions evolve over time by combining and mutating the best ones until an optimal result is found

[8] Whole numbers

DTs offer several advantages over CTRNNs. First, they produce discrete outputs at each time step, making them ideal for controlling musical events like notes, while CTRNNs require discretization of their continuous outputs. Additionally, DTs are easier to analyze and manipulate, making their decision-making process more transparent than the opaque nature of CTRNNs [9]. DTs are also more computationally efficient, executing fewer conditional operations, and their structure allows them to evolve more easily. They can start simple and grow by adding nodes, and they can adapt in real time by adjusting thresholds for decision-making.

The second study used genetic algorithms to evolve DTs with specific dynamic properties, like maximizing the number of leaf nodes[10] visited, to create an active and responsive system. Although the DTs were evolved in an abstract environment, they showed responsiveness to real audio input during testing.

In a generative music system, the DTs controlled both discrete and continuous parameters. Discrete outputs could select specific sound samples, while the internal state modulated playback parameters. The system also featured an interactive grid that allowed DT decisions to trigger combinations of sound events, creating a flexible mapping between the DT's behavior and the musical output (22).

Bown, who used both the DT and CTRNN systems found the DT-based system more responsive and easier to collaborate with when designing interactive compositions. The study also noted a "shadowing" effect with DTs, where sudden changes in audio input would reliably trigger changes in the DT's output, enhancing the system's interactivity.

Both studies in Bown's paper explore the potential of live algorithms to drive generative

[9] Or any Neural Networks if given enough neurons in this case

[10] The endpoints of a decision tree where no further splits occur, and they represent the final outcome or decision

music systems in a modular and creative manner. While CTRNNs offer rich, continuous musical behaviors, they are more challenging to control and understand compared to the more predictable and discrete outputs of DTs. Bown's paper concludes that a modular approach to live algorithm design, with different algorithms suited to different musical tasks, is a productive direction for future research and creative exploration in live musical performance (22).

3.2 Google's Magenta(1)

Google's Magenta is an open-source research project designed to explore the role of machine learning and artificial intelligence in creative tasks, particularly in generating art and music. It was trained on a mix of public domain music and music that was licensed for training, in other words, therefore, it is all under fair use. Magenta builds on the capabilities of deep learning models, employing neural networks to assist in creating new and original musical compositions, artwork, and other creative outputs. The project aims to push the boundaries of creativity by developing tools that allow both professionals and amateurs to engage with AI in the creative process. Magenta's music generation relies on deep neural networks, including RNN (with LSTMs, page 24) and Variational Autoencoders (VAEs) (1), to understand musical structures and generate coherent and stylistically consistent music.

Magenta's core function in music generation revolves around modeling sequences, which are key to music's temporal nature. Music, by its essence, is a time-based art form where each note or rhythm depends on what came before. To capture this sequential structure, Magenta employs RNNs, which are particularly well-suited for handling sequence data like melodies or rhythms. RNNs have a memory mechanism that allows them to retain information from previous time steps, enabling the model to generate music that not only progresses naturally but also maintains coherence over longer compositions. These networks are trained on vast datasets of existing music to learn patterns, progressions, and relationships between notes and chords. After training, the RNNs can generate music by predicting the next note in a sequence based on the previous ones, making the process somewhat similar to how human musicians compose by recalling musical patterns and themes.

A significant challenge in generating music is creating long sequences that remain coherent and musically satisfying over time. Magenta addresses this by using an advanced variation of RNNs known as LSTM networks (page 24.) LSTMs are designed to overcome the limitations of traditional RNNs by better capturing long-term dependencies, allowing the model to generate compositions that make sense across longer time spans. For instance, LSTMs help ensure that a melody introduced early in a piece is revisited or resolved later, creating a more musically cohesive structure. This memory component is essential for producing compositions that are not just random sequences of notes but rather structured pieces with a beginning, middle, and end.

In addition to RNNs, Magenta also uses VAEs (page 28,) which play a different role in the music generation process. VAEs are a type of generative model that can learn a compressed, abstract representation of input data and then generate new data from this learned representation(9). The process starts with the encoder, which takes the input data and compresses it into a latent representation. Unlike a traditional autoencoder, where this latent representation is a fixed point, in a VAE it is described by a probability distribution—typically a Gaussian.[11] The encoder outputs two parameters: the mean and the variance [12]. These parameters define a Gaussian distribution in the latent space.

[11] A bell-shaped probability distribution used to represent the latent space, helping the model learn smooth and continuous representations of the input data

[12] Or more commonly, the logarithm of the variance

Once the latent distribution is defined, the VAE introduces a sampling step, where it draws a sample from this distribution. This is where the model adds randomness, allowing for the generation of new data points later on. To make this step differentiable [13], VAEs employ a technique called the reparameterization trick. Instead of sampling directly from the distribution defined by the mean and variance, the model samples from a standard Gaussian distribution and then shifts and scales this sample using the mean and variance parameters. After the sample is drawn from the latent space, the decoder takes this sample and tries to reconstruct the original data. The decoder is trained to learn a mapping from the latent space back to the original data distribution. The training objective of a VAE involves two parts. The first part is the reconstruction loss, which measures how well the decoded output matches the original input data. The second part is a regularization term, often called the KL [14] divergence, which ensures that the learned latent distribution[15] is close to a standard normal distribution. This regularization helps the VAE avoid overfitting and encourages it to learn a smooth, continuous latent space where nearby points generate similar outputs.

VAEs allow Magenta to explore variations on existing musical ideas by capturing high-level features such as melody, harmony, or rhythm and then generating new compositions based on these features. For example, VAEs can take a musical theme and create multiple variations of it, each unique yet sharing the same underlying structure. This capability is particularly useful in creative tasks where variation and experimentation are key.(9)

One of Magenta's standout features is its ability to generate music that aligns with specific user inputs or constraints. Users can input a short musical idea or melody, and the model will generate a continuation or accompaniment based on that input. This interaction between

[13] So that back propagation can be used for training

[14] Kullback-Leibler

[15] A simplified representation of the data, where the model captures important features in a lower-dimensional space, typically following a Gaussian (normal) distribution

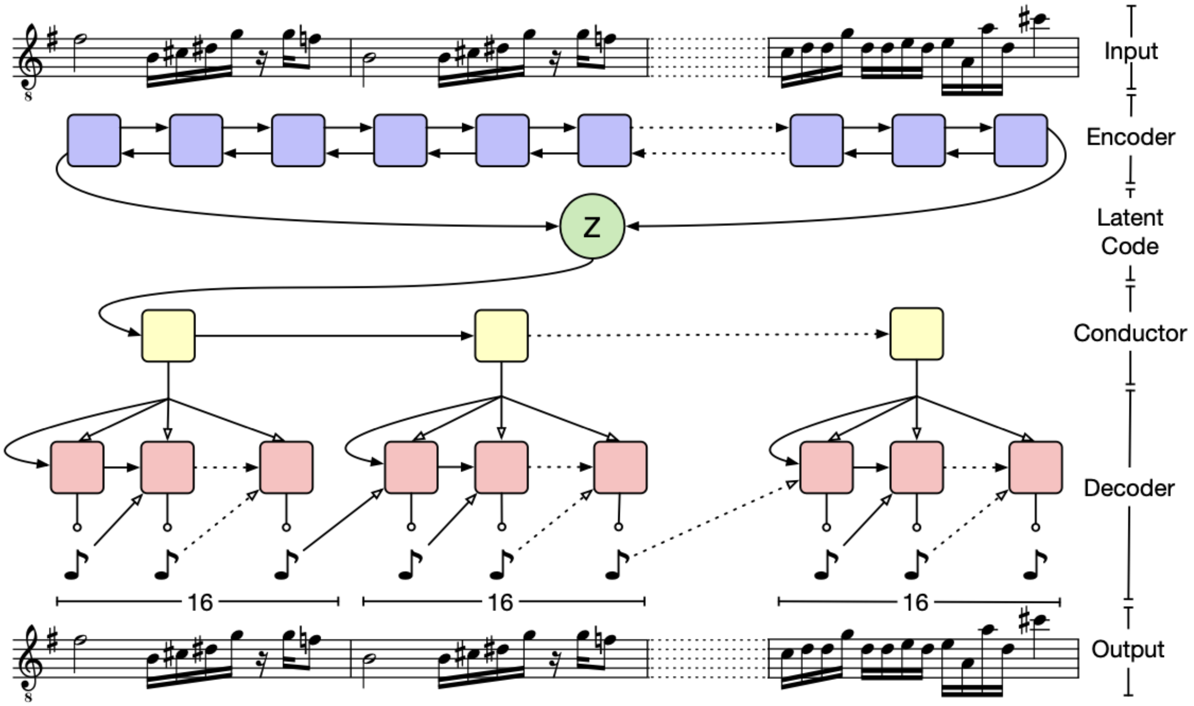


Figure 3.1: An illustration on the encoding and decoding of music with MusicVAE(9)

human and machine allows users to guide the AI's output while still benefiting from the generative power of the model. This aspect of Magenta makes it not just a tool for generating random music, but hypothetically a collaborative system that enhances human creativity. When used properly and trained ethically, it could provide musicians and composers with a means of exploring musical ideas that they might not have conceived of independently, expanding their creative possibilities.

Magenta's user interface and tool sets are designed to be accessible to a wide range of users, from professional musicians to hobbyists and researchers. The project offers several pre-trained models that can be used out of the box, along with a flexible framework for training custom models on specific datasets. This flexibility makes Magenta a versatile tool for anyone interested in experimenting with AI-generated music. By providing both ready-to-use tools and the ability to customize models, Magenta bridges the gap between machine learning research and practical creative applications.

However, despite its impressive capabilities, Magenta does face certain limitations. One of the main challenges is generating music that captures the emotional depth and nuance typically found in human-composed music. While Magenta can generate compositions that are stylistically consistent and musically coherent, the subtleties of human emotion, intention, and creative intuition remain difficult for AI models to replicate. Additionally, because Magenta relies on existing musical data to train its models, the system is constrained by the limitations of its training data. This means that while Magenta can interpolate between different styles of music, its ability to create entirely novel styles or transcend existing musical conventions is still limited.

Magenta also requires significant computational resources to train and generate music, particularly for more complex tasks involving large datasets or high-resolution audio. This can pose a barrier to entry for users without access to advanced hardware. However, Google has worked to make Magenta more accessible by providing cloud-based tools and frameworks, allowing users to leverage its capabilities without needing to invest in expensive infrastructure.

3.3 Open AI's Jukebox(2)

OpenAI's Jukebox is an advanced deep learning model designed to generate music, complete with vocals, instrumental tracks, and lyrics, in a wide variety of genres and styles. Like Magenta, Jukebox is a non-commercial research project designed to advance scientific understanding of music generation. It is open-source. The code and model weights are available online, allowing researchers and developers to explore and utilize the model for music generation tasks. Jukebox builds on the success of neural network models in tasks such as image generation and text processing, adapting these approaches to the unique challenges of music generation. In essence, Jukebox employs a combination of hierarchical generation and conditioning on metadata to produce music that is coherent, stylistically accurate, and highly detailed, overcoming some of the limitations that earlier music generation models faced. (2)

Jukebox functions through a hierarchical approach to music generation, which operates at multiple levels of temporal resolution(2). This hierarchical process is central to Jukebox's ability to generate coherent music over extended durations. At the highest level of the hierarchy, Jukebox outlines the broader structure of the piece, such as the chord progression, melody, and general style across several bars of music. This level provides a global framework, defining the direction of the music in terms of key musical elements like harmony and form. Once this larger structure is established, the model then fills in finer details at lower levels, such as individual notes, rhythms, and timbres, ensuring that the piece remains consistent and musically logical. This multilevel approach enables Jukebox to generate music that does not simply loop or repeat but evolves dynamically over time, maintaining coherence throughout the piece(2).

A distinguishing feature of Jukebox is its ability to be conditioned on metadata. Metadata refers to contextual information such as the genre, the artist style, and even specific lyrics. By conditioning the model on these variables, Jukebox can generate music that adheres to a specified genre or imitates the style of a particular artist, making the generated music feel authentic and faithful to the user's input. Moreover, the conditioning on lyrics allows the system to generate vocal tracks that match the lyrics, ensuring that the vocal melody aligns rhythmically and thematically with the lyrics provided. This capacity to condition the model on diverse forms of metadata makes Jukebox a versatile and powerful tool for generating a

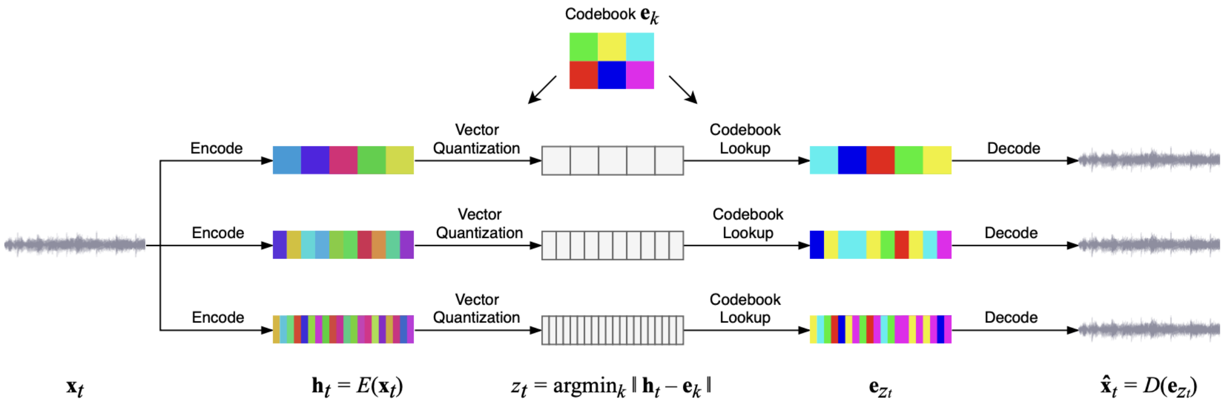


Figure 3.2: Jukebox's hierarchical architecture (2)

wide range of musical styles and pieces that not only sound stylistically accurate but also exhibit thematic coherence.

Jukebox's technical backbone consists of a complex deep learning model architecture, with particular reliance on autoregressive models and VAEs(page 28.) An autoregressive model predicts the next element in a sequence based on prior elements, which is a critical feature for generating sequences like music. Music is inherently sequential, with each note and rhythm depending on what came before, and autoregressive models are well-suited to capturing this structure. In Jukebox, this model is applied hierarchically to generate music progressively, first defining the broader aspects of the piece and then refining the details.

One of the primary challenges in generating music, particularly with vocals, is the alignment of different musical components, such as the melody, harmony, and rhythm, with the lyrics. Jukebox addresses this challenge by employing a method called upsampling, which refers to generating higher-resolution versions of a low-resolution input. In the context of Jukebox, the model first generates a low-resolution version of the music, including the vocals, and then iteratively refines this version, adding more details and increasing the fidelity of the music with each step. This process ensures that all elements of the music—vocals,

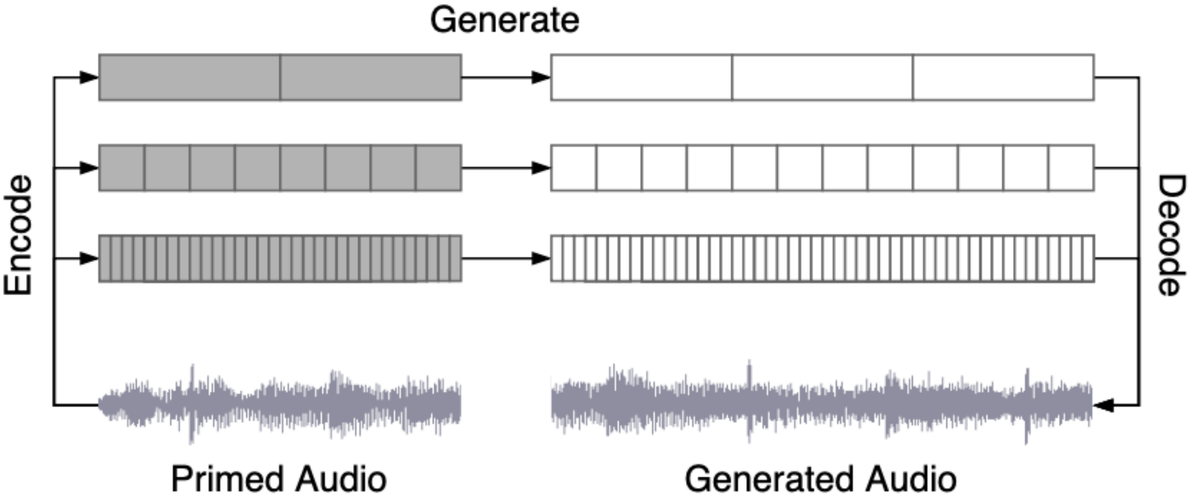


Figure 3.3: Jukebox's sampling method(2)

instrumentals, and rhythms—are aligned properly and sound natural together.

Despite its sophistication, Jukebox still faces certain limitations. One of the most notable is that while the model is capable of generating impressive musical pieces, it requires substantial computational resources, and the quality of the generated music, while high, may not yet reach the level of professional human composers in terms of complexity and emotional depth.[16] Additionally, like Magenta, the model's reliance on large datasets of pre-existing music means that it can only generate music within the styles it has been trained on, and while it can interpolate between these styles, it is less adept at producing entirely novel forms of music that fall outside of its training data. Hypothetically, there is a chance that the model might occasionally produce music in novel forms by combining/evolving from the styles on which it has been trained.

Moreover, while Jukebox's hierarchical model can generate coherent music over long peri-

[16] The ability to evoke complex feelings and connect with listeners on a profound, personal level through its melody, harmony, dynamics, and other expressive qualities, e.g. intentional chromatism

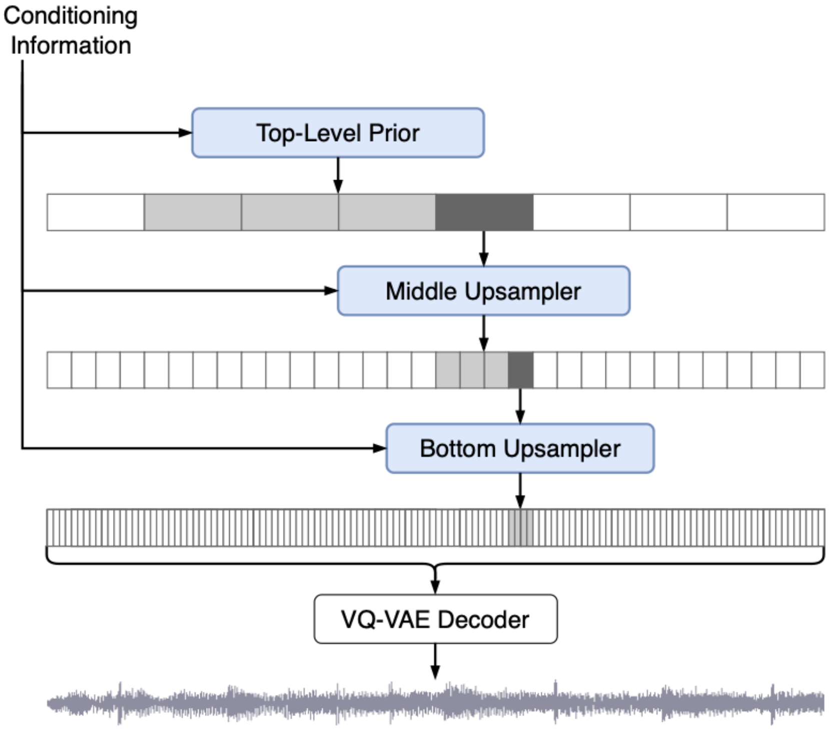


Figure 3.4: Jukebox's upsampling at inference(2)

ods (about five minutes,) its ability to maintain structure over pieces longer than five minutes remains a challenge. Music often relies on subtle variations and long-term development, and although Jukebox can handle short-term coherence, it struggles with creating music that evolves organically over longer durations, like a symphony or extended composition.

One potential problem if one desires to use Jukebox for music generation, even if it is not for commercial use, is that the specific details regarding the training data used for Jukebox have not been publicly disclosed. Given the nature of the project and its non-commercial intent, the training data likely included a mix of public domain works and other music, though explicit information about licensing and authorizations has not been provided.

3.4 MuseGAN(3)

MuseGAN is a novel approach for generating multi-track symbolic music[17] using GANs (page 24.) It is an open-source project. The code and resources are publicly available online, allowing researchers and developers to access and contribute to the project. Unlike traditional methods that often focus on generating single-track monophonic music, MuseGAN targets polyphonic, multi-track music generation, where different instruments play simultaneously. This requires models that can handle the complexity of multiple musical parts working together, with each part unfolding in time yet being interdependent on others. It also focuses on generating music in a symbolic form, such as MIDI-like piano-roll representations, where the temporal progression and interaction of notes across multiple instruments or tracks are critical.

At its core, MuseGAN employs GANs, which consist of two networks: a generator and a discriminator, which compete with each other in a game-like setting. The generator creates fake data[18] by sampling from a random noise distribution. Its goal is to produce data that is as realistic as possible. The discriminator receives both real data[19] and fake data[20], and its job is to distinguish between the two. It assigns a probability indicating how likely the input data is real or fake.

During training, the generator learns to create better fakes to fool the discriminator, while the discriminator improves at telling the difference between real and fake data. This adversarial process continues until the generator becomes so good that the discriminator can no longer distinguish between real and fake data reliably. This makes GANs powerful for

[17] MIDI files or sheet music

[18] Like images, text, or music

[19] From a training dataset

[20] From the generator

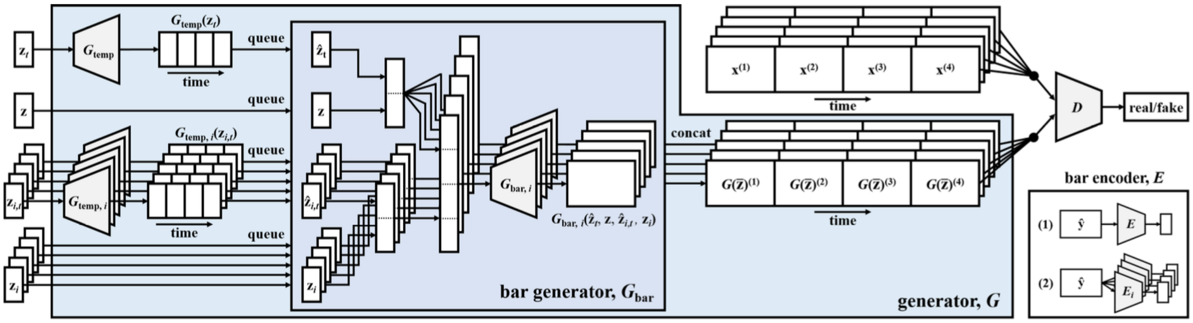


Figure 3.5: MuseGAN's multi-track GANs architecture(3)

generating realistic data, such as images, text, and music.

In this case, the generator learns to produce music, while the discriminator evaluates how realistic the generated music is by comparing it to real music. The generator's goal is to fool the discriminator into thinking that the music it generates is real, and the discriminator's goal is to correctly distinguish between real and generated music. Through this adversarial process, both networks improve over time, with the generator producing increasingly convincing music.

MuseGAN introduces several variations of its model to handle the complexity of multi-track music. One version, called the jamming model, features multiple independent generators, each responsible for creating a specific track, such as drums, bass, guitar, and piano. Each generator operates independently, and each track is evaluated by a separate discriminator. This approach allows for a high degree of autonomy between the tracks but can sometimes result in disjointed or uncoordinated music, as each track is generated separately.

In contrast, the composer model uses a single generator to create all tracks simultaneously. This generator is guided by a single random input, which can be viewed as representing the composer's overall intention. The resulting multi-track music is evaluated collectively by a single discriminator. This approach ensures greater coherence between tracks, as they are generated together with an understanding of how each part relates to the others.

MuseGAN also introduces a hybrid model that blends elements of both the jamming and composer models. In this setup, each track is generated by a separate generator, but these generators share some inputs, allowing them to coordinate and produce more harmonious music. A single discriminator evaluates the overall quality of the generated music. This approach allows for flexibility in the generation process, enabling individual tracks to have some independence while still maintaining inter-track coherence.

To capture the temporal structure of music, MuseGAN extends beyond the bar-level generation used in simpler models. In its "generation from scratch" method, it generates multiple bars of music by using two sub-networks: one that learns the overall temporal structure and another that generates the individual bars of music. This approach ensures that the music has continuity and progression, as the generator can learn how musical ideas evolve over time.

For interactive applications where a human composer might want to provide one or more tracks, MuseGAN also supports "track-conditional generation." In this setup, the generator is conditioned on one or more existing tracks, such as a melody or a chord progression, and it learns to generate the remaining tracks to complement the given input. This can be particularly useful for generating musical accompaniments or for collaborative music composition with AI.

The model represents music in a multi-track piano-roll format, where time is represented on the horizontal axis, and pitch is represented on the vertical axis. A note is indicated by a mark on the piano-roll, and each instrument or track has its own piano-roll. This allows it to generate polyphonic music, where multiple notes and instruments are played at the same time.

To evaluate the quality of the generated music, MuseGAN uses several objective metrics that measure aspects like harmonic coherence, rhythm, and inter-track dependency. The model is trained on a large dataset of symbolic music, such as the Lakh MIDI Dataset, and evaluated both through these metrics and through subjective listening tests. These tests show that MuseGAN is capable of generating music that is not only musically coherent but also pleasant to listen to, with different versions of the model excelling at different aspects of music generation.

In the context of MuseGAN, for example, the need for handling temporal dependencies in music can theoretically be partially addressed by integrating GRUs into its architecture[21], MuseGAN focuses on polyphonic, multi-track music generation, and while its primary mechanism is the GAN framework, the temporal structure of music could be further improved by introducing GRUs to handle dependencies across time, especially in scenarios where harmonic progression and rhythmic patterns evolve over long sequences.

However, like Jukebox, MuseGAN's training data are not all in the public domain, and explicit authorizations from original authors may not have been obtained.

3.5 IBM's Watson Beat

IBM's Watson Beat is another open-source project, it integrates various machine learning methodologies to support music composition in partnership with human artists. Though technical specifics of Watson Beat are not as publicly detailed as other AI systems, it is clear that the system relies on several key approaches. One major component is cognitive computing, a core feature of IBM's Watson platform. Cognitive computing allows Watson Beat to simulate human thought processes, applying music theory, structure, and emotional content to produce compositions that are both technically sound and emotionally engaging.

[21] Although it doesn't, but it's a thought

22

Machine learning algorithms play a central role, enabling Watson Beat to analyze large datasets of music from different genres. By learning the patterns, structures, and defining elements of these genres, the system generates new music that fits the style or mood specified by the user. While the exact architecture of Watson Beat is not disclosed, it is likely that neural networks, especially recurrent RNNs or Long LSTM networks, are employed. These types of networks are well-suited for sequential data like music, as they can retain information over time, allowing for coherent and fluid compositions.

Watson Beat also allows for user interaction, where users can specify parameters such as mood, genre, or instrumentation. This interactive feature ensures that the system tailors the generated music to the user's input. The user input system may use decision trees or rule-based algorithms to map these inputs to specific musical outcomes.

A key characteristic of Watson Beat is its emphasis on collaboration between AI and human musicians. The system functions as a creative partner rather than a replacement for human composers. It enables bidirectional interaction, allowing human artists to influence the AI's output and edit the AI-generated content, ensuring the final product aligns with the artist's creative vision. However, the specific details regarding the training data used for Watson Beat have not been publicly disclosed. Consequently, it's unclear whether the training data comprises public domain works or if explicit authorizations from original authors were obtained. If trained ethically for future commercial use, Watson Beat could serve as a tool that enhances the creative process, providing new opportunities for expression and experimentation while working alongside human composers.

The four aforementioned models are intended to be scientific research projects and, there-

[22] Though everyone has a different definition, this generally refers to a piece's ability to deeply resonate with listeners, evoking feelings, memories, or a strong connection

fore, are not designed for commercial use. They are all open-source projects, granting researchers and developers access to their code to further advance this technology.

3.6 Commercialization of modern models

With the boom in modern music generation attempts, follows the commercialization of said models. More and more services become available for the everyday consumers. Services like Suno.ai, Udio, and Musicfy not only spark everyday users but also controversy. The root cause lies within the training process of generative AI models like this. This is not an isolated issue with music, in fact, creative works across all fields are actively fighting back against this process called "scraping". Scraping is defined by the action of AI companies "downloading" the internet and feeding the works of others to their machine learning models, training them to create by studying and analyzing the styles of existing artworks without the consent of the original creators. This in turn hurts the livelihood of said creators, since now users can simply pay the AI companies for their services instead of commissioning the original creators even when they're going after a certain style that is known as that particular artists'. On the one hand, one could argue that this makes art and music accessible to everyone, but on the other hand, it is done at the expense of someone's hard work being stolen. Some AI advocates would say that the models learn in a fashion that is not unlike how a person would learn, for example, as composer, you listen to tons of music, and you learn by "copying" certain style. But AI has the computing power far superior to human abilities, making the playing field incredibly uneven. One example would be Midjourney[23] allowing its user to enter prompts such as "create the character design in the style of Hayao Miyazaki," then accurately producing the image in this style.

[23] An AI image generation service

In June 2024, major record labels (Universal Music group, Sony Music Entertainment, and Warner Music group) have sued Suno.ai and Udio AI companies for allegedly stealing copyrighted materials to generate music(23). Without disclosing how they have trained their AI model, Suno.ai released a statement saying the model does not work by memorizing and regurgitating pre-existing content, instead, it's designed to generate completely new outputs since their technology is transformative[24]. They also state that their "mission is to make it possible for everyone to make music."(24)

Since then, a lot of the generative AI services have disabled the feature that allowed their users to mimic a certain artist in hopes of avoiding potential copyright infringement. Although styles cannot be copyrighted, text prompts that include an artist's name might cause commercial models to generate outputs that plagiarize specific pieces of music by the artist on which they were trained. The problem, however, is not going away anytime soon, since legislators are slow to adapt to a digital world that is rapidly evolving ever since the birth of Open AI's ChatGPT back in 2022. Many artists and musicians have already been scraped against their wills, and there still aren't laws established for royalties from scraping, let alone an "opt out" option for the artists. In a juncture like this, one is faced with a dilemma, should one take their music off the internet in an age where online presence plays a major role especially for artists that are just starting up in order to protect their intellectual properties, or should one look for ways to compete with the new technologies, so music written by human doesn't become obsolete too soon. This is by no means an easy question to answer, and creative workers have to understand that, technological advancement will not stop because a group of people do not like it. Humans create arts in order to express, make a statement, and ultimately evoke resonance, with AI that perspective of art making is lost. However, that is not to deny the functionality of AI tools. If there must be a war in the near

[24] The word "transformative" is used to describe how their model works

future over artistic autonomy and rights, it will not be between AI and humans, it will be the companies that wield the powers of AI that are coming after the human artists. Machine learning algorithms by themselves are merely tools, whether to take or to give up the power is entirely up to us. Sun Tzu says in “The Art of War” that, “If you know the enemy and know yourself, you need not fear the result of a hundred battles.”(25) It may not be our enemy (although many creative workers may think it is,) the purpose of this dissertation is to start to understand the tool that is generative AI, in hopes of using it as a tool to aid composers in the composition process. Note that this is not to say that we as composers should hand our composition process over to AI, but to use the AI as a digital assistant to handle tasks like orchestration and etc.

3.7 What does it all mean?

The generative AI tools explored in this dissertation are inherently dependent on the data used to train the AI music generation models, as their functionality is entirely contingent upon the patterns, structures, and stylistic features embedded within their datasets. These models do not possess an innate understanding of music; rather, their "musical brains" are constructed through the curation, preprocessing, and feeding of data. In essence, their ability to generate music is bound by the statistical relationships and patterns present in the data, limiting their capacity to innovate or create beyond the confines of their training.

This dependency underscores a central challenge in developing AI for music composition: how to teach the unknowable. Music, as a deeply human form of expression, often transcends tangible rules or codified structures. It is shaped by cultural, emotional, and experiential contexts that defy straightforward quantification. Teaching AI to "understand" these ineffable aspects involves approximations rather than direct transmission of knowledge. For instance, embedding metadata, such as emotional intent or cultural context, into training datasets allows models to simulate understanding, but these simulations remain rooted in statistical likelihoods rather than genuine comprehension. Similarly, iterative fine-tuning, where a model's outputs are adjusted through human feedback, creates heuristics for emulating expressiveness or stylistic fidelity without the AI truly grasping the underlying intent.

The development of an AI model's "musical brain" involves a complex process of representation learning, where relationships between musical elements are encoded into abstract mathematical spaces. Through hierarchical modeling, advanced systems begin with broad musical structures—such as key, tempo, and form—before filling in details like melodic lines, harmonic progressions, and dynamics. This process mirrors, to some extent, the layered approach of human composition, but it lacks the lived experience and creative intuition that underpin human artistry. Instead, the AI relies on algorithmic processes, such as gradient descent and back propagation, to optimize its ability to generate music that adheres to the stylistic tendencies and structural rules present in its training data.

This leads to the philosophical question of whether these tools can truly "know" music. Their outputs can evoke emotional responses and demonstrate stylistic coherence, but their knowledge is statistical rather than experiential. The creative leaps that characterize human artistry—those moments of inspiration driven by personal experience, cultural influence, and emotional resonance—remain inaccessible to AI. The AI's "creativity" is thus combinatorial, assembling elements from its training data into novel but ultimately derivative configurations.

The implications of this contingency are profound, particularly in an era where creative works are increasingly subject to practices like data scraping. The AI's reliance on training data raises ethical and professional concerns about intellectual property, artistic agency, and the commodification of creativity. These tools, while powerful, are not autonomous entities; they are shaped by the data and frameworks provided by their human developers. As such, the role of the composer shifts from creator to collaborator and curator, guiding these systems to produce outputs that align with artistic goals while addressing the limitations and ethical challenges inherent in their use.

CHAPTER 4

The hunt for a model

Thus, the quest to find a modern, easy to understand, easy to use, and easily accessible music generative AI model begins. My first choice was Google's Magenta, since it is open-source and readily available. However, by the time I acquired the skills to install it, Google had moved forward with Ableton, and have made it into a plugin called Magenta Studio for Ableton Live. Although I do not have Ableton Live, I found the web-based demo provided by Google on their TestKitchen. Upon testing the model, it produces somewhat satisfactory results. However, since the generated excerpt is short and there is no information about the training data, it is difficult to make a meaningful comparison with other types of models, such as autoregressive models.[1]

4.1 MusicLM(4)

The first model I encountered was MusicLM – a sophisticated model designed for generating high-fidelity music from text descriptions. The system builds upon prior advancements in audio generation, specifically leveraging the framework established by AudioLM, which models audio in a hierarchical, autoregressive manner. However, MusicLM extends this by incorporating a significant feature: text conditioning. This allows the model to not only generate audio but to produce music that corresponds directly to descriptive text inputs,

[1] Based on open-source access

such as "a calming piano melody with soft background vocals."

The methodology employed by MusicLM is hierarchical in nature, which means that the model generates music across multiple stages, each adding progressively finer details to the audio. Initially, the model generates semantic tokens, which capture the high-level structure of the music, such as melody and rhythm. This ensures that the broader aspects of the music, such as the overall mood and genre, adhere to the textual description provided. In the second stage, these semantic tokens are transformed into acoustic tokens, which represent the finer details of the sound, such as timbre and sound quality. This hierarchical method allows MusicLM to generate music that is both coherent over extended periods (up to five minutes) and rich in detail, addressing the challenge of maintaining long-term consistency in generated audio.

To train the model, MusicLM relies heavily on SoundStream and MuLan. SoundStream provides the audio tokenization framework, allowing for high-fidelity audio compression and reconstruction. MuLan, on the other hand, facilitates the text conditioning aspect of the model. It is a joint music-text embedding system that learns representations for both music and text in the same space, ensuring that the generated music aligns closely with the text prompt. During training, MusicLM uses MuLan embeddings derived from audio data, while during inference, it uses MuLan embeddings computed from the input text descriptions.

One of the major challenges in developing a system like MusicLM is the scarcity of paired music-text datasets. To overcome this, the model uses MuLan embeddings from large-scale, audio-only datasets during training. This approach eliminates the need for paired data and allows MusicLM to learn from vast amounts of music without requiring corresponding text annotations. Moreover, the researchers behind MusicLM introduced a new dataset called MusicCaps, which contains 5.5k high-quality music-text pairs curated by expert musicians. This dataset plays a crucial role in evaluating MusicLM's performance and improving its

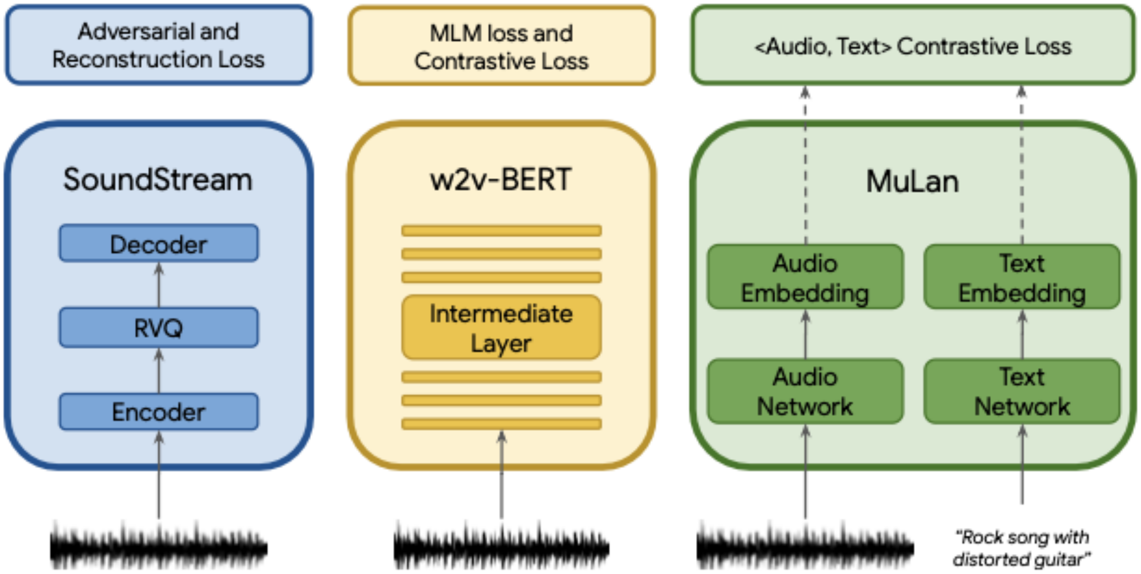

Adversarial and Reconstruction Loss
MLM loss and Contrastive Loss
<Audio, Text> Contrastive Loss
SoundStream
Decoder
RVQ
Encoder
w2v-BERT
Intermediate Layer
MuLan
Audio Embedding
Text Embedding
Audio Network
Text Network
"Rock song with distorted guitar"

Figure 4.1: MusicLM encoding input using SoundStream and MuLan (4)

ability to generate music that aligns with text prompts.

The model's architecture is based on decoder-only Transformer model,[2] and it uses temperature sampling[3] to generate diverse yet coherent music sequences. The model can also generate longer musical pieces by autoregressively predicting subsequent tokens based on previous ones, which enables it to produce coherent music over extended periods. Additionally, MusicLM introduces a feature called "story mode," which allows for dynamic changes in the text prompt over time, generating music that evolves with changing descriptions.

While my dissertation discusses the contributions of RNNs and LSTMs to sequence generation, it is important to acknowledge the transformative impact of transformer architectures,

[2] A type of machine learning model designed to process and understand sequential data, like text, by using self-attention mechanisms to focus on the most relevant parts of the input

[3] A method used in AI generation to control how random or focused the output is, with higher temperatures making it more creative and lower temperatures making it more predictable

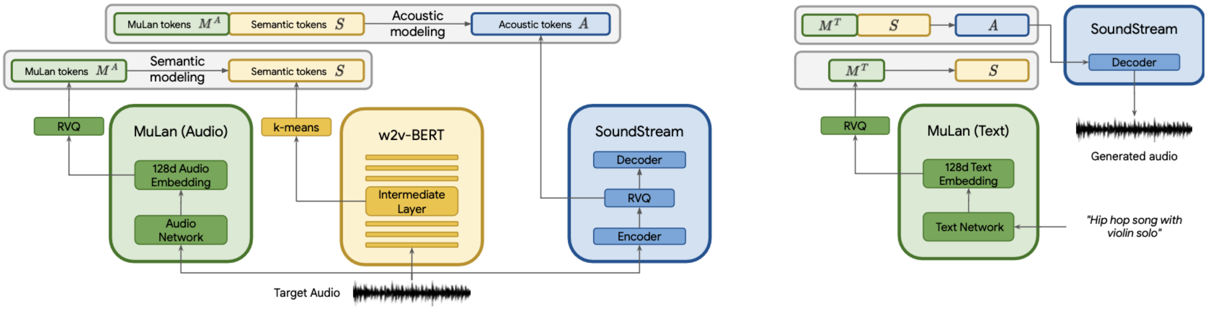


Figure 4.2: Training (left) and inference (right) (4)

which have emerged as the state-of-the-art in handling sequential data. Transformers, first introduced in the "Attention Is All You Need" paper(26), represent a significant advancement by replacing the recurrent nature of RNNs and LSTMs with self-attention mechanisms. This shift allows them to process long sequences more efficiently by attending to relevant portions of the input data without the limitations of sequential memory. In the context of music generation, transformer-based models like MusicLM leverage these capabilities to generate complex, stylistically coherent outputs across extended durations. The inclusion of transformers in these tools aligns with the same architecture that powers models like ChatGPT(27), illustrating their versatility across domains. By improving computational efficiency and contextual understanding, transformers have paved the way for advancements in generative AI, including the tools and methods explored in this dissertation.

In terms of evaluation, MusicLM significantly outperforms its predecessors, such as Mubert and Riffusion, both in audio quality and faithfulness to the input text descriptions. This is measured using various metrics, including the Frechet Audio Distance (FAD), which evaluates audio quality, and the MuLan Cycle Consistency (MCC), which assesses how closely the generated music adheres to the text prompt. The model also underwent human evaluations, where listeners preferred MusicLM's generated samples over other systems.

4.2 EnCodec(5) and MusicGen(6)

After the setback with MusicLM, the search for a model that runs locally continues. Luckily, I came across an autogressive model called MusicGen(6). MusicGen is a model that, like all other models discussed in this paper, is open-source. Similar to Magenta, it is trained on a combination of music in the public domain and licensed music. It introduces a novel approach to music generation that integrates high-quality output with user control, utilizing text descriptions or melodic inputs to direct the creative process. The system addresses a critical challenge in the field: generating music that not only exhibits coherence but also aligns with specific user-defined parameters such as genre, instrumentation, or melodic structure. It presents a streamlined architecture that simplifies the generation process while maintaining a high degree of flexibility and precision.

But before we dive into MusicGen's methodologies, we must first understand the codec model that it is built on – EnCodec(5). EnCodec is a state-of-the-art neural audio compression model introduced by Meta AI's FAIR team, designed to deliver real-time, high-fidelity audio at lower bit rates. EnCodec uses a deep learning-based encoder-decoder architecture to compress audio signals efficiently while preserving their perceptual quality. This model significantly improves on traditional audio compression methods by leveraging neural networks to represent audio signals more compactly and reconstruct them with minimal artifacts. The model has been trained to handle various audio formats, including speech and music, while maintaining real-time processing capabilities, making it suitable for streaming applications.

At the heart of EnCodec is its encoder-decoder system. The encoder compresses the audio signal into a latent representation, which is then quantized to produce a compact, discrete representation of the audio. This quantized data is then passed to the decoder, which reconstructs the original audio from this compressed format. One of the key features

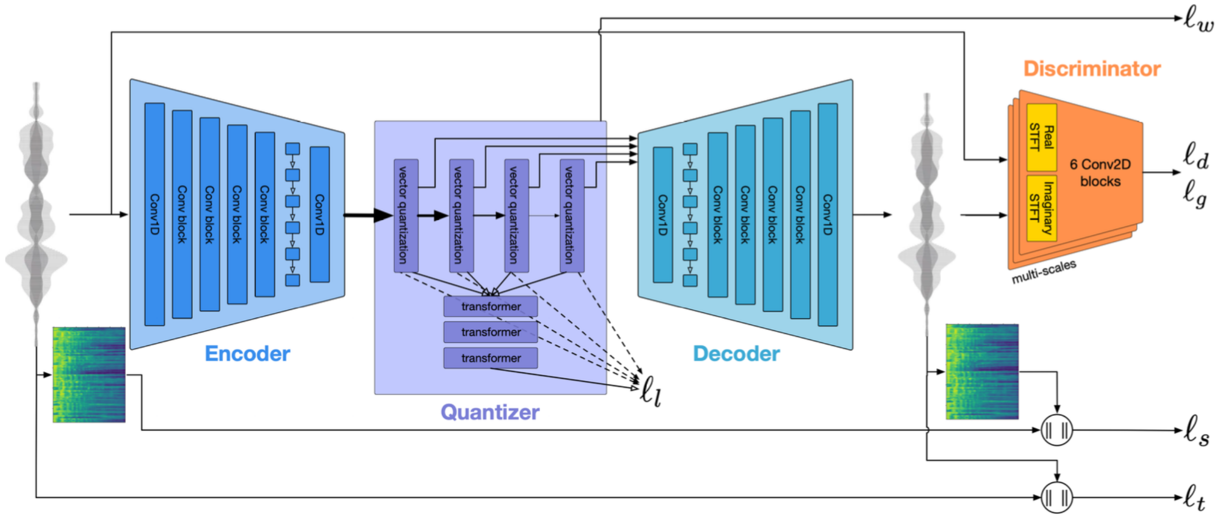


Figure 4.3: EnCodec's encoder decoder system for music compression(5)

of EnCodec is the use of Residual Vector Quantization (RVQ), a technique that refines the quantization process by successively applying multiple quantization steps. This helps the system achieve high levels of compression without losing significant audio quality. RVQ allows EnCodec to support various bit rates, adjusting the number of quantization steps dynamically to meet different bandwidth constraints.

EnCodec also employs multi-scale adversarial loss functions, which improve the perceptual quality of the reconstructed audio by focusing on different time-frequency representations of the audio signal. These loss functions ensure that the model not only minimizes the error between the original and compressed audio but also reduces perceptual distortions, producing higher-quality audio that sounds more natural to human listeners. Additionally, a unique loss balancer is used to stabilize the training process by ensuring that each component of the model contributes proportionately to the overall learning process.

A notable advantage of EnCodec is its ability to handle real-time audio processing. The model is designed to be computationally efficient, running on a single CPU core while still processing audio faster than real time. This is achieved through careful architectural design and optimization, including the use of lightweight Transformer models for entropy coding.[4] The Transformer component compresses the quantized representation even further by modeling the structure of the compressed audio data, allowing for additional reductions in bandwidth without sacrificing audio quality. Moreover, because of the addition of the Transformer models, it is able to skip the reverse RVQ (page 50) step and go directly to the decoder, which significantly reduces the inference[5] time. This makes EnCodec particularly well-suited for music streaming, where low-latency, high-fidelity audio compression is critical.

EnCodec is versatile in terms of the audio formats it can handle. It supports both mono and stereo audio at different sample rates (24 kHz and 48 kHz), and it is trained on a variety of audio domains, including clean speech, noisy speech, and music. The model has been evaluated extensively using both objective metrics[6] and subjective evaluations[7] to compare its performance against traditional codecs like Opus and EVS, as well as other neural audio codecs like Lyra-v2. Across these evaluations, EnCodec consistently outperforms its competitors, particularly at lower bit rates, where maintaining high perceptual quality is challenging.

One of the innovations in EnCodec is the use of variable bit rate training, which allows the model to operate at multiple bit rates ranging from 1.5 kbps to 24 kbps. This flexibility enables it to adapt to different bandwidth constraints dynamically, providing high-quality audio even at lower bit rates. The model's use of entropy coding, combined with the Trans-

[4] A data compression technique that reduces the size of data by representing frequently occurring items with shorter codes and less common items with longer codes

[5] Generating output after training

[6] Such as signal-to-noise ratio

[7] MUSHRA tests

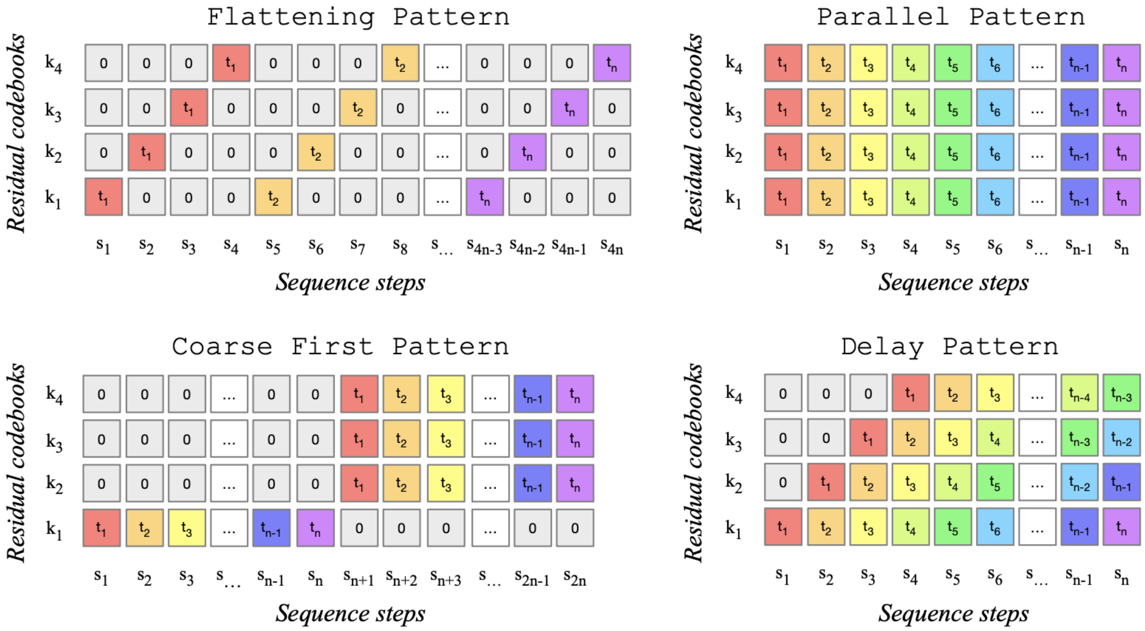
former model, further enhances its ability to compress audio efficiently while keeping the processing time within real-time limits. With EnCodec being the foundation of MusicGen, the researchers took the models trained on EnCodec's datasets, as while as taking the RVQ, Transformer, and Decoder architectures to build MusicGen. Central to MusicGen's methodology is a single-stage transformer-based language model. Transformers have demonstrated significant success in tasks involving sequential data processing, such as language and audio. In MusicGen, the model operates on a set of compressed discrete music representations, commonly referred to as tokens, which are derived from audio data. Unlike previous systems that necessitated multiple stages or hierarchical models for music generation, MusicGen employs a single-stage model to manage the entire process from start to finish. This simplification enhances the system's efficiency and scalability, enabling more widespread use without compromising the quality of the generated music.

A fundamental challenge in music generation, as opposed to speech, is the increased complexity of the audio signal. Music generally involves higher sampling rates than speech due to its wider frequency spectrum. For example, while speech models might operate at a sample rate of 16 kHz, music typically requires 44.1 kHz or 48 kHz, thereby increasing the amount of data the model must process. Moreover, music contains intricate harmonic and melodic relationships that must be carefully preserved during generation. Even minor errors in harmony or rhythm can be jarring to listeners, rendering accuracy paramount. MusicGen addresses this issue by utilizing a system of compressed tokens, which represent the music in a more manageable, quantized format while still capturing the essential details required for high-fidelity output. These tokens are generated using a RVQ, a core feature of the EnCodec model employed in MusicGen. RVQ compresses the audio into discrete tokens, enabling the transformer model to process them while maintaining a balance between detail and computational efficiency. Each token stream represents different levels of the audio's detail, capturing the nuances of the music in a compressed format. By decomposing the audio into these multiple streams, MusicGen can model the dependencies between different components of the music, ensuring that the generated audio remains coherent and musically rich.

An important aspect of MusicGen is its arrangement of these token streams for processing. The model employs a technique called codebook interleaving, which organizes the tokens in a specific sequence to facilitate processing by the transformer. Various methods exist for arranging these tokens, and the choice of pattern affects both the model's complexity and the quality of the generated music. In MusicGen, the researchers experimented with several patterns, ultimately determining that the "delay" pattern achieves an optimal balance between computational efficiency and the preservation of audio quality. This pattern introduces a small delay between the different token streams, allowing the model to process them more efficiently without sacrificing the fine-grained details crucial to music generation. Imagine the token streams make up a choir, while the Parallel Pattern represents the choir singing homophonically, causing the subsequent instances to copy the differences caused by the RVQ discretizing the temporal latent space vectors (decoder outputs.) The Delay Model minimizes the losses by delaying each token stream generated by each codebook, so not all losses are carried over to the next instance. This is like having the token stream choir "sing" a canon.[8]

A key feature of MusicGen is its capacity to allow users to control the music generation process through text and melody conditioning. Text conditioning enables users to input descriptive phrases, such as "an upbeat rock song with electric guitar," which guide the model in creating music that aligns with the given description. To achieve this, the system

[8] A composition where one melody is played and then repeated by another voice or instrument, starting at different times but overlapping, like a musical round



Flattening Pattern

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	$s_{\dots}$	s_{4n-3}	s_{4n-2}	s_{4n-1}	s_{4n}
k_4	0	0	0	t_1	0	0	0	t_2	...	0	0	0	t_n
k_3	0	0	t_1	0	0	0	t_2	0	...	0	0	t_n	0
k_2	0	t_1	0	0	0	t_2	0	0	...	0	t_n	0	0
k_1	t_1	0	0	0	t_2	0	0	0	...	t_n	0	0	0

Parallel Pattern

	s_1	s_2	s_3	s_4	s_5	s_6	$s_{\dots}$	s_{n-1}	s_n
k_4	t_1	t_2	t_3	t_4	t_5	t_6	...	t_{n-1}	t_n
k_3	t_1	t_2	t_3	t_4	t_5	t_6	...	t_{n-1}	t_n
k_2	t_1	t_2	t_3	t_4	t_5	t_6	...	t_{n-1}	t_n
k_1	t_1	t_2	t_3	t_4	t_5	t_6	...	t_{n-1}	t_n

Coarse First Pattern

	s_1	s_2	s_3	$s_{\dots}$	s_{n-1}	s_n	s_{n+1}	s_{n+2}	s_{n+3}	$s_{\dots}$	s_{2n-1}	s_{2n}
k_4	0	0	0	...	0	0	t_1	t_2	t_3	...	t_{n-1}	t_n
k_3	0	0	0	...	0	0	t_1	t_2	t_3	...	t_{n-1}	t_n
k_2	0	0	0	...	0	0	t_1	t_2	t_3	...	t_{n-1}	t_n
k_1	t_1	t_2	t_3	...	t_{n-1}	t_n	0	0	0	...	0	0

Delay Pattern

	s_1	s_2	s_3	s_4	s_5	s_6	$s_{\dots}$	s_{n-1}	s_n
k_4	0	0	0	t_1	t_2	t_3	...	t_{n-4}	t_{n-3}
k_3	0	0	t_1	t_2	t_3	t_4	...	t_{n-3}	t_{n-2}
k_2	0	t_1	t_2	t_3	t_4	t_5	...	t_{n-2}	t_{n-1}
k_1	t_1	t_2	t_3	t_4	t_5	t_6	...	t_{n-1}	t_n

Figure 4.4: Different types of output patterns (6)

converts the text input into a format that can be understood by the transformer model. The researchers experimented with different text encoding methods, including T5, FLAN-T5, and CLAP, to map the text input into a form that effectively conditions the model's output.

Melody conditioning provides an additional layer of control by allowing users to input a specific melody, which the model uses as the foundation for the generated music. This feature is particularly useful for musicians or composers who have a specific melodic idea in mind but wish the model to develop it into a full piece. To implement melody conditioning, MusicGen uses a chromagram[9] representation, which captures the harmonic structure of the input melody. This chromagram is then input into the model as a conditioning signal, guiding the generation process to ensure that the resulting music aligns with the provided melody. Notably, MusicGen does not require labeled data for this process, enhancing its

[9] A way to represent spectral audio information as a visual representation of pitch

flexibility and accessibility.

The results of MusicGen are noteworthy. The system outperforms other state-of-the-art models, such as Riffusion and Mousai, in both objective measures—including FAD, Kullback-Leibler Divergence, and CLAP Score—and subjective listening tests, where human evaluators rated the quality and relevance of the generated music. MusicGen is capable of generating music at a sampling rate of 32 kHz, which, while slightly lower than the typical 44.1 kHz of standard music production, still produces music of sufficiently high quality to satisfy most listeners.

One of the critical findings in the research is the effectiveness of the codebook interleaving patterns. Through ablation studies,[10] the researchers demonstrated that selecting the appropriate interleaving pattern [11]is crucial for balancing the trade-offs between model complexity and audio quality. The Delay Pattern was found to be particularly effective, as it reduces computational load while maintaining the quality of the music. This balance allows the model to generate music more efficiently, making it feasible for a broader range of users, from casual creators to professional musicians.

Beyond its technical achievements, MusicGen has broader implications for the field of music generation. Its simplified architecture could make music generation more accessible to a wider audience, including individuals without extensive technical expertise. The ability to control the output through text and melody conditioning makes it a valuable tool for musicians, composers, and producers, who can use it to explore new creative possibilities. Moreover, the system's design allows for future improvements in fine-grained control[12] over

[10] Systematically removing or modifying parts of the model or its features to understand their impact on the quality and behavior of the generated music

[11] The pattern that organizes the tokens in a specific sequence to facilitate processing by the transformer

[12] The ability to precisely adjust specific aspects of the generated music, such as melody, rhythm, or style, to achieve a desired outcome

the generation process, potentially unlocking even greater expressive potential.

Despite its strengths, the authors and developers of MusicGen acknowledge that MusicGen has limitations, particularly in achieving fine-grained control over certain aspects of the music. The model relies on broad conditioning signals, such as text or melody, but does not yet offer the ability to control more detailed musical elements like individual instrument tracks or specific rhythmic patterns. Future research could explore more sophisticated conditioning methods and expand the diversity of the datasets used for training. A possibility is combining the MusicGen with a hierarchical architecture, making it suitable for generating music with clearer structures. The authors also address the ethical considerations of using large-scale generative models in music, particularly concerning issues of data bias and the potential for AI-generated music to compete with human artists.

CHAPTER 5

Implementation

To adapt the MusicGen model to generate music that aligns with my musical style, we implemented a systematic fine-tuning process. This process involved enhancing the pre-trained model with a dataset of my original compositions. By employing this approach, we were able to leverage the foundational capabilities of the pre-trained MusicGen model while tailoring it to generate music that reflects my unique compositional style. In the following sections, I will provide a detailed description of the computational methods employed in this work to fine-tune the MusicGen model on Google Colab, a cloud computing platform. The training process utilized a NVIDIA T4 Tensor GPU with 12.0 GB of GPU RAM. The process had to take place on Google Colab due to the technical constraints posed by my equipment, but if another composer were to attempt the same thing with a computer with slightly better specs, the whole thing can be done locally. The only potential obstacle would be the inaccessibility of the GPU posed by the MacOS system.

5.1 Getting the model ready

5.1.1 Dataset preparation

Our initial step involved the collection of a diverse set of training music, which I composed. To create this dataset, I first organized a collection of both newly composed and past works that are representative of my compositional style. The collection contains about 2 hours of original music, in genres such as film score, musical, film song, and EDM. To ensure consistency and compatibility with the MusicGen model, all audio files were exported with a uniform sampling rate of 44.1 kHz. A uniform sampling rate can prevent data misalignment, which can lead to errors during training and hence unwanted degraded performance. This conversion was crucial for maintaining alignment with the model's input specifications, facilitating seamless integration during the subsequent training stages.

5.1.2 Data preprocessing

Following data collection, we preprocessed our training data by dividing each music file into segments of 30 seconds. This segment length was selected based on the architectural limitations of the model (MusicGen with 300M parameters). To further refine the training dataset, we organized the music samples into training, validation, and testing subsets. Each file was meticulously labeled with descriptive tags capturing the essence and stylistic nuances of the piece, providing contextual guidance to the model throughout the fine-tuning phase.

5.1.3 Loading and configuring the pre-trained MusicGen model

To initiate the fine-tuning process, we employed PyTorch on Google Colab to load the pre-trained MusicGen model. This platform was selected to harness the advanced computational resources available in a cloud environment, thereby facilitating the efficient processing of the dataset. To safeguard data security and privacy, the training data are loaded from Google Drive, which is not legally accessible to other commercially available AI companies like Suno, and the output model at each checkpoint is saved to Google Drive as well.

5.1.4 Model fine-tuning

In the fine-tuning process, we employed an adaptive moment estimation optimizer (AdamW) in conjunction with categorical cross-entropy[1] as the loss function.[2] This configuration has proven effective in similar generative tasks. AdamW's stability in convergence, resilience to hyperparameter sensitivity,[3] and effective balance between exploration and exploitation make it an optimal optimizer for tasks requiring precise fine-tuning, such as music generation. It enables robust learning and generalization by dynamically adjusting the learning rate for each parameter. Categorical cross-entropy, on the other hand, aids the model in learning a probability distribution over potential outcomes, enabling it to produce outputs that align with specific musical patterns or styles. This combination optimized the model's learning trajectory, allowing it to adapt more accurately to my compositions. In the context of deep learning, iterations and epochs are terms related to the training process. An iteration is a single step in training when the model processes a fraction of the entire training set, whereas an epoch refers to one complete pass (a thousand iterations) through the entire training set, consistently monitoring performance across both the training and validation sets to assess convergence and employ an early-stopping method to prevent overfitting. We conducted four trials of the fine-tuning process, with the first trial containing only a quarter, and five epochs of the entire data set to ensure the viability of the method, while the other three trials were done with the complete data set but different amounts of epochs.

[1] A measure of how well the model predicts the next note or musical event, with lower values indicating more accurate predictions

[2] A mathematical tool that measures how far the generated music is from the desired output, helping the model learn and improve during training

[3] How changes in the model's settings, like learning rate or batch size, affect its performance and the quality of the generated music

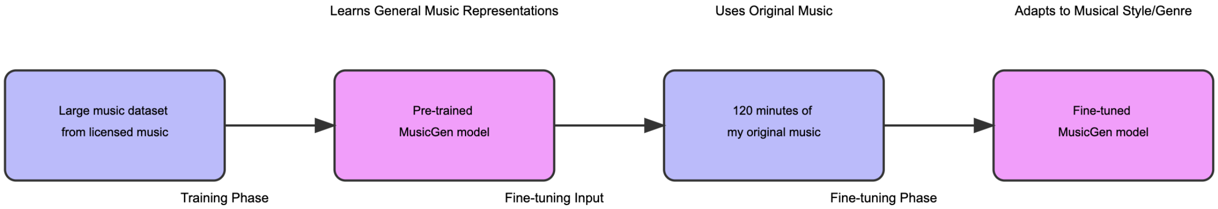


Figure 5.1: Simplified fine-tuning process

5.1.5 Training the fine-tuned Model

During the training phase, we observed a gradual decrease in the categorical cross-entropy (CE) loss and perplexity (PPL) values across epochs, indicating that the model is progressively learning from the training data, suggesting that the model is effectively adapting to the data. The gradient norms initially exhibit high values and exhibit some initial instability but stabilize in later epochs, reflecting a more consistent and controlled update process as training progresses. A gradual learning rate decay throughout the training process facilitates the model's convergence, thereby avoiding large parameter updates in later stages. This is crucial for producing a stable fine-tuned model. We periodically save checkpoints, particularly when a new best state for validation loss is achieved. This tracking of the most performant model state during training ensures that, even if subsequent epochs exhibit a decline in performance, the most effective model can be retained.

5.2 Evaluation of the fine-tuned model

To assess the model's performance, we used PPL (perplexity) as the primary metric, as it serves as an indicator of a model's predictive accuracy. In music generation, lower perplexity values indicate that the model has successfully acquired the ability to produce outputs that closely align with the training data. All four of our trials achieved satisfactory perplexity scores, this data suggests the model's capacity to generate music that aligns with the stylistic attributes of my original compositions. One thing worth pointing out before we delve into the details of each trial is that the attempts that had successfully generated outputs all had a temperature of "1", meaning the model was allowed to be "creative" and deviate from the provided text prompt, while a "0" would mean for the model to adhere strictly to the prompt. When given temperature below "1", the model would only generate outputs of silent audio files.

5.2.1 Trial one

We first tested the water with trial one, running five epochs (5,000 iterations,) to fine tune the model with approximately 30 minutes of music. At the end of epoch one, the model achieved a CE of 2.228 and a PPL of 35.907, reflecting its initial performance. By epoch two, these metrics improved to a CE of 1.551 and a PPL of 6.566, indicating significant learning progress. epoch three showed further refinement, with a CE of 1.222 and a PPL of 4.284. The improvement continued in epoch four, achieving a CE of 1.094 and a PPL of 3.611. Finally, by the end of epoch five, the model reached a CE of 0.998 and a PPL of 3.237, marking consistent progress and suggesting the model became increasingly adept at generating sequences with reduced uncertainty and higher accuracy. These results demonstrate effective fine-tuning, with steady decreases in both CE and PPL across epochs.

With a final PPL of 3.237, it was apparent that the model had been fine-tuned to the provided data set, and there was a possibility of some overtraining since the data set size was small; the generated materials should reflect the stable state of the model. With the first trial, the main goal was to have a model that would generate music regardless of the style to determine the viability of the methodology. The resulting model demonstrated somewhat promising results, with music that features little to no influence from the data set. For example, when given a prompt that includes labels such as “orchestral,” “acoustic guitar,” and “and cello,” the model generated something that sounded quite absurd. However, at least this step proved this method to be viable.

5.2.2 Trial two

The fine-tuning process again demonstrates a clear downward trend in CE and PPL values across successive ten epochs, reflecting the model’s improving performance. Initially, the CE starts at 5.234 and drops to 3.112 by the final epoch, while PPL begins at 187.36 and decreases significantly to 22.46, underscoring the model’s enhanced ability to predict and encode data patterns. The gradient norms stabilize around an average of 0.125, suggesting a well-behaved optimization process throughout training. Validation metrics align closely with training results, with the validation CE decreasing from 5.523 to 3.218 and the validation PPL reducing from 205.74 to 25.36. These consistent reductions in both training and validation metrics underscore the success of the fine-tuning process in optimizing the model’s performance.

However, the final PPL is not ideal due to the model not being trained enough, the resulting model generates music that somewhat resembles my musical style, but should the text prompts resemble the labels provided in the dataset too much, it would generate something that it too close to the music in the dataset, implying a potential overfitting issue.

5.2.3 Trial three

Due to trial two’s higher than trial one value of PPL, we decided to increase the number of epochs to fifteen, adding 5,000 additional iterations to train the model with the same data set. In epoch one, CE started at 3.369 and PPL at 64.032, which decreased to 2.645 and

26.226 by the end of the epoch. Progress continued in epoch two with CE at 2.185 and PPL at 14.791, and further improved in epoch three with CE of 2.044 and PPL of 12.701. By epoch five, the CE had dropped to 1.859 and PPL to 9.871, showing steady adaptation. This trend persisted through the next epochs, with CE reaching 1.608 and PPL 7.257 by epoch eight. By epoch ten, CE was at 1.512 and PPL at 6.834. The model continued refining, ending epoch fifteen with a CE of 1.379 and a PPL of 6.243.

Although a clear downward trend exists in both CE and PPL values, the resulting model was more unstable than the last trial. It is a hit or miss, with a success rate of about 50 percent, meaning half of the time, it would not generate music at all when given a prompt. Though the reasons are uncertain, our educated guess is that during training, the model has encountered "catastrophic forgetting" (also known as catastrophic inference). It is a phenomenon observed in artificial neural networks during sequential learning tasks, particularly during fine-tuning. It occurs when a model trained on a new dataset or task experiences a significant degradation in its ability to perform previously learned tasks. This issue arises because the neural network's parameters are overwritten during training on the new data, erasing or diminishing the knowledge encoded for the original tasks. Catastrophic forgetting is especially problematic when a pre-trained model is adapted to a new, often narrower, domain. Pre-trained models such as MusicGen are trained on large-scale datasets to capture a wide range of features and patterns. Fine-tuning involves updating the model to specialize in a specific task or dataset. However, without proper precautions, the updates can overwrite the general-purpose knowledge learned during pre-training, losing the model's broad applicability.

5.2.4 Trial four

Because of the turn of event in trial three, we decided to fine tune the model again, this time running only five epochs to see if we could avoid the catastrophic forgetting that happened in the last trial. In the first epoch, the CE starts at 3.864 with a PPL of 161.216, and by the end of the epoch, the CE decreases to 3.291, corresponding to a PPL of 59.292. This trend continues into the second epoch, where the CE reduces further to 2.707 and the PPL to 25.365. By the third epoch, the CE drops to 2.420, and the PPL to 18.485. The fourth epoch shows even better results, with a CE of 2.206 and a PPL of 14.388. Finally, in the fifth epoch, the CE reaches 2.025 and the PPL decreases to 12.056, demonstrating the model's progressive refinement and the increasing alignment of predictions with the training data. Validation summaries echo this improvement, with CE decreasing from 3.746 in epoch one to 3.169 in epoch five, and PPL declining from 42.366 to 23.772. The resulting model is even more unstable than that of trial three, when prompted, the model often would not generate outputs that contained any actual music, sometimes it would output what can be described as "sound effects" at best, they are sparse and unpredictable.

5.2.5 Findings and thoughts

One interesting observation is that out of all the trials, the attempts that do yield successful results are the ones without any prompts. They often sound more like the pieces we provided in the fine-tuning dataset. The presumption here is the differences in labeling. While we have a system of labeling the musical pieces in the dataset, these labels might have different meanings in the original data used to train the pre-trained model, whether how the developers labeled their data or how the model categorized the data. Because the fine-tuned model is built upon the pre-trained model, the conflicting labels might have confused the model, making it unsure how to execute some of the keywords we provided in the text prompts.

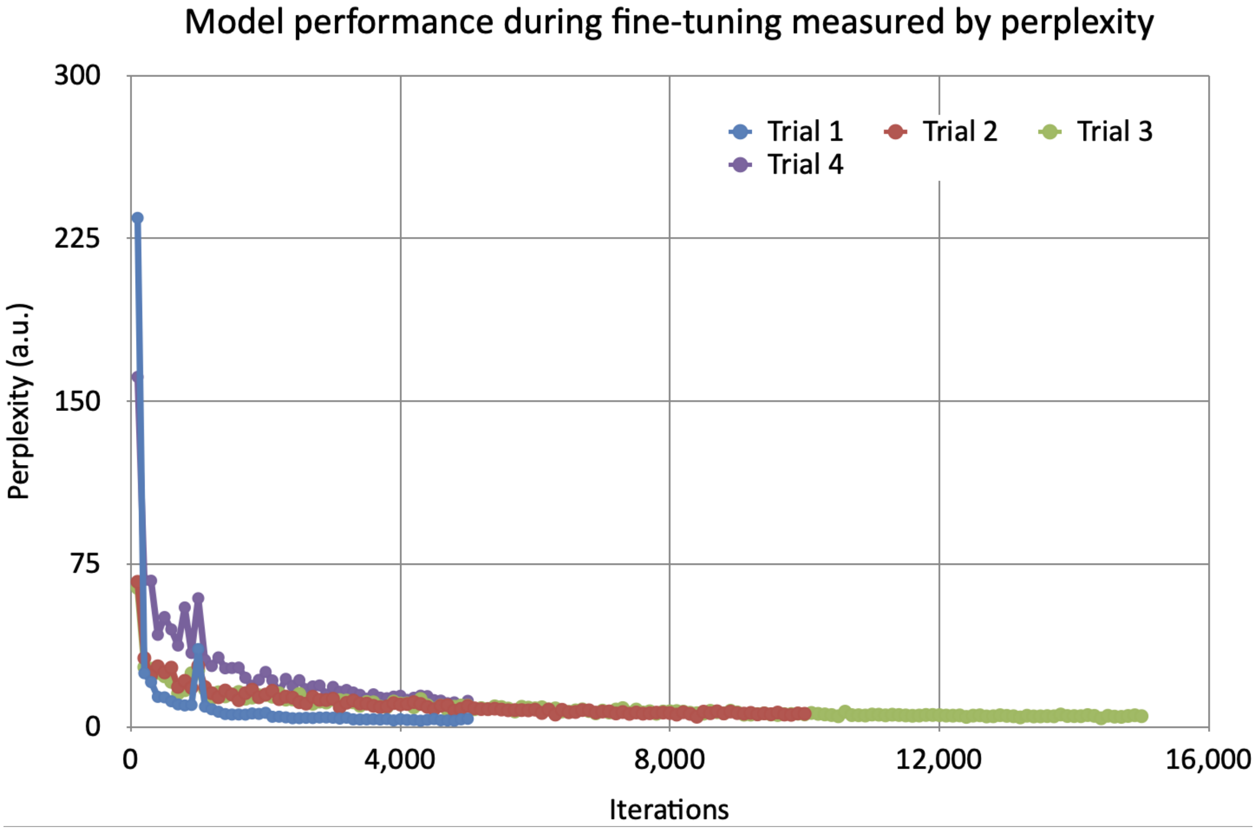


Figure 5.2: PPL trends for Trials 1 through 4

Through this fine-tuning process, we fine-tuned the MusicGen model to produce music that somewhat mirrors my compositional style, though not so much harmonically speaking; certain orchestral traits from the dataset are apparent in the outputs. By adhering to the aforementioned process and leveraging both the pre-trained model's capabilities and a dataset of my original compositions, we created a model that generates musical outputs that align with some of our specific artistic preferences.

However, that is not to say that the fine-tuned models do not have issues, on the contrary one immediate problem that stems from the outputs is the lack of structure, both micro and macro. AI models like MusicGen operate on token-level sequence prediction, treating musical elements as discrete units without explicitly addressing their theoretical relationships.

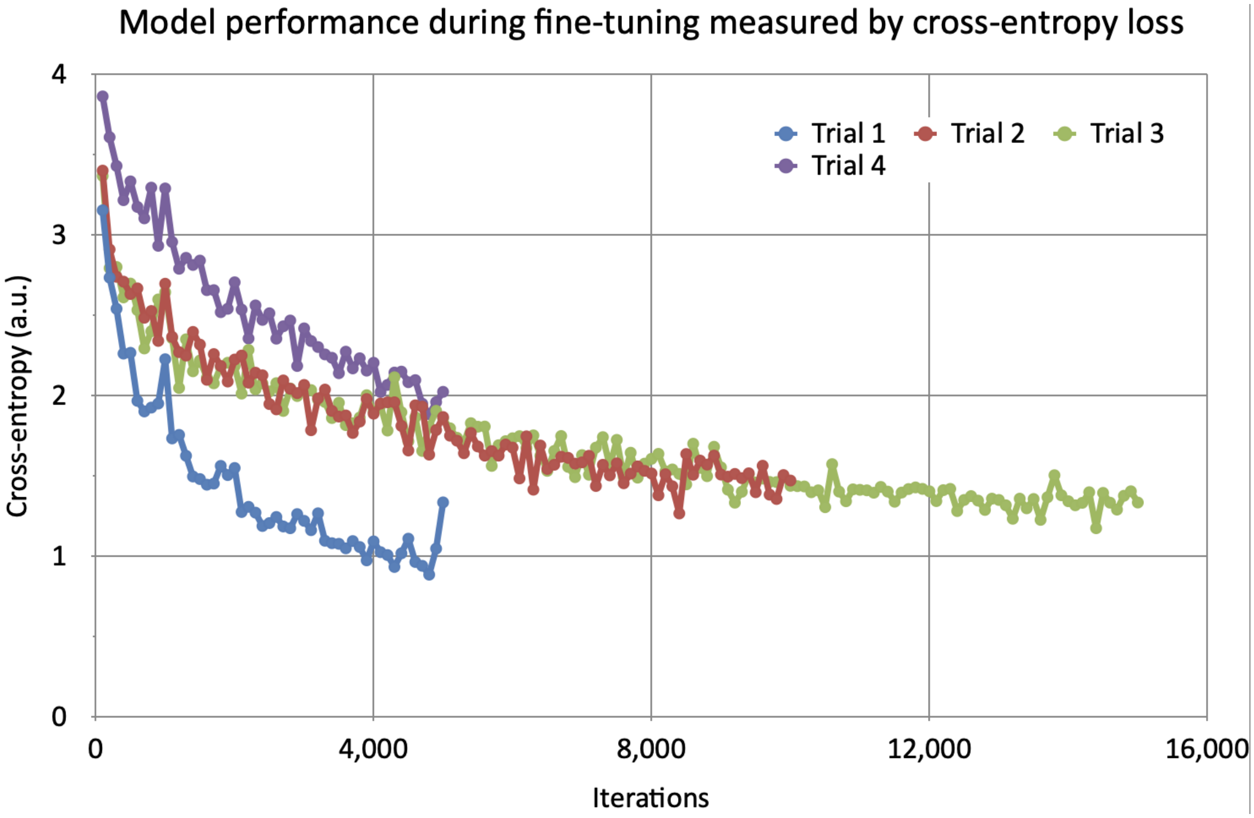


Figure 5.3: CE trends for Trials 1 through 4

While this approach allows for stylistic diversity, it often results in compositions that lack coherent harmonic progressions, voice leading, and phrase-level structural integrity. The lack of explicit representation for concepts such as tonality, chord functions, and cadences hinders the model's ability to generate music that aligns with Western theoretical frameworks. The core limitation of models like MusicGen lies in their reliance on statistical learning. These models generate music by predicting sequences based on probabilities derived from training datasets. While this allows them to replicate patterns of harmonic progressions observed in their training data, they lack an understanding of why these progressions occur or their theoretical underpinnings. This disconnect arises from several factors.

First, the models operate as black boxes, focusing on mapping input to output without explicit rule-based logic. Second, their training data, while vast, is not annotated with the functional roles or theoretical structures of harmony. This lack of explicit labels prevents the model from differentiating between statistical regularities and music theory principles. Lastly, the models emphasize local coherence, generating short-term sequences that may sound plausible but fail to exhibit a global harmonic structure. Integrating harmonic understanding into AI music models poses significant challenges. Traditional music theory relies on rule-based systems, which are deterministic and do not align easily with the probabilistic nature of deep learning. Moreover, encoding music theory into a neural network would require extensive annotation of training datasets to include functional harmonic roles, such as tonic, dominant, or subdominant, as well as the relationships between them. Balancing this integration while maintaining the generative flexibility of current models is a complex task.

One solution would be building a hierarchical architecture for the model. A hierarchical architecture divides music generation into interconnected levels, each responsible for different aspects of the composition process. This layered approach introduces structured decision-making and allows for explicit integration of theoretical principles at various stages. By segmenting tasks across hierarchical layers, the model can simultaneously address global and local musical aspects. A top layer might handle overarching structures such as key centers and harmonic progressions, while intermediate layers focus on melodic and harmonic interplay within these constraints. A lower layer could refine details like note-level phrasing and stylistic ornamentation, ensuring theoretical adherence and expressive coherence.

The hierarchical framework allows for the integration of Western music theory at different stages of the generation process. The top layer could establish tonal centers, harmonic progressions, and large-scale modulations. Intermediate layers would be tasked with generating voice leading, secondary harmonies, and phrase-level dynamics, ensuring that these elements align with the theoretical structure established in the top layer. Finally, the bottom layer would handle embellishments and ornamentation, ensuring note-level decisions are consistent with the broader harmonic and stylistic context. This layered approach allows the model to maintain both theoretical coherence and creative fluidity, producing music that resonates with listeners.

One of the key advantages of a hierarchical design is its ability to preserve context across multiple levels of composition. Local decisions, such as individual chord choices or melodic phrasing, can be informed by broader structural goals, such as tension and resolution across a movement. This ensures that harmonic choices and thematic developments align with the overarching framework of the composition, producing music that feels intentional and cohesive.

A hierarchical model could achieve significant improvements in areas critical to Western music, such as cadences, voice leading, and thematic development. It would enable the generation of functional harmonic progressions, including authentic cadences that define phrase closures and establish resolution. Voice leading would benefit from smooth transitions between chordal voices, reducing dissonance and enhancing overall cohesion. Thematic motifs could be effectively guided through their introduction, evolution, and recapitulation, ensuring structural integrity and emotional impact across sections of the composition.

Despite its promise, the hierarchical approach presents certain challenges. It introduces additional computational complexity and requires training data that reflect detailed theoretical structures. Balancing rigid adherence to music theory with creative flexibility also poses a significant challenge, as overly rigid systems may stifle innovation. Nonetheless, advancements in hierarchical modeling hold great potential for transforming AI music generation, allowing these systems to produce compositions that are both statistically and theoretically coherent.

5.3 Scoring with MusicGen

Although the test results are not what I had envisioned before, I had to put the now fine-tuned model into action; after all, the purpose of this paper is to find a feasible way for an independent creative worker to work with a locally run fine-tuned generative AI model. Some of the cues in my dissertation piece are composed with this model. During the process, I first prompt the model with text, including keywords that define the genre, instrumentation, and general mood that I am trying to achieve, and cross my fingers and hope that it would infer something of use.

Out of the countless numbers of tries, I have struck gold with trial two's model three times. The first time is the result of a prompt that says "menacing drone," the output is a drone that has no particular harmonic movement but interesting colors and timbres. Therefore, there was no way to transcribe the output generated by the model. The drone was then imported into a Digital Audio Workstation (DAW[4]) session file, where I initially attempted to incorporate it as part of the orchestration. After further listening, I realized the need to separate the mid-range and lower frequencies for more versatile use. To achieve this, I considered two approaches: duplicating the track and applying equalizers to isolate the desired frequency ranges, or utilizing a stem-splitting tool to separate the frequencies directly from the audio file. I opted for the latter.

As the process progressed, I found myself desiring greater control over individual pitches, which would enable more harmonic possibilities beyond the static nature of the drone. To address this, I bounced the bass and mid-frequency outputs as separate files and used them in a granular synthesizer, effectively transforming them into new instruments for cues throughout *Infinite*. This cue required extensive modifications to the model's raw output due to

[4] Examples include Logic Pro, ProTools, and Ableton Live

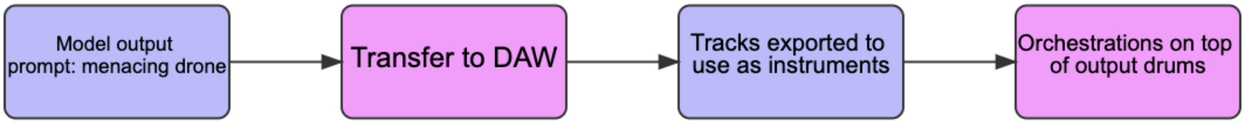


Figure 5.4: The workflow of cue 1 of *Infinite*

its lack of rhythmic or harmonic movement. Ultimately, the processed output functioned as a supportive instrumental element rather than the primary focus of the composition.

The second time is the result of the prompt “hip-hop” after numerous unsuccessful attempts using more elaborate prompts containing descriptors like "upbeat" or "laid-back." This simplified input led the model to generate a beat resembling a conventional hip-hop rhythm, which was subsequently used to rescore cue 7 of *Infinite*. In addition to the percussion track, the output included other elements, such as a piano and a bassline, though these were not immediately aurally distinct. The model’s limited understanding of harmonic structure resulted in a repetitive bassline characteristic of hip-hop music, a feature that was both a strength and a limitation in this context.

The generated audio file was then imported into the DAW session file, with the tempo set to adapt for Logic Pro’s tempo analysis. Though it may not sound like it, the tempo is actually constantly changing. The original output audio file was split into three separate tracks—drums, piano, and bass— by the Logic Pro stem splitter for further processing. The percussion track remained unaltered throughout the cue, as it effectively provided a rhythmic foundation that aligned with the desired aesthetic. Minimal transcription was required for this cue due to the suitability of the drum track in establishing a basic rhythmic structure.

However, the repetitive nature of the generated bassline and piano tracks necessitated selective muting in certain sections. To introduce greater harmonic and orchestral flexibility, new basslines were composed and orchestrated around in specific parts of the cue. Portions of the piano track were incorporated into the final product without modification, as its "lo-

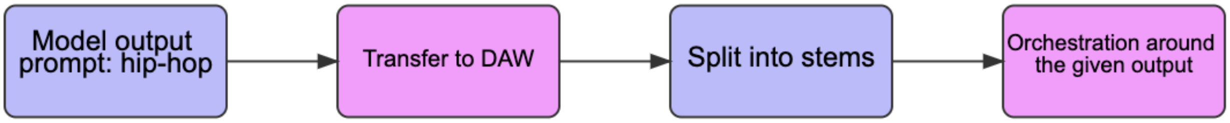

Model output prompt: hip-hop
Transfer to DAW
Split into stems
Orchestration around the given output

Figure 5.5: The workflow of cue 7 of *Infinite*

fi" timbre contributed a desired texture and character to the overall sound. This process exemplifies the balancing act between leveraging the strengths of AI-generated material and addressing its limitations through creative intervention, ultimately shaping the output into a cohesive composition that meets the needs of the cue.

The third time is the most surprising: using the simple prompt "piano," the model generated a piece featuring a piano part with both a melody and accompaniment. Unlike previous outputs, this piece exhibited harmonic motion and a relatively steady sense of tempo, though it lacked the nuanced logic typically associated with tonal harmonies as traditionally understood. In its raw form, the output was overly "cheesy" for the delicate emotional tone required for cue 3. Consequently, further modifications were necessary to adapt the material to suit the scene appropriately.

After transferring the raw audio file into the Digital Audio Workstation, the tempo was pre-set to adapt for better synchronization during subsequent processing. An attempt was made to separate the track into stems to gain control over the melody and accompaniment individually. However, this approach proved unsuccessful due to the overlapping frequency ranges of both parts. As a result, the process required aural transcription of the model's output, followed by a quick harmonic analysis to identify and interpret the underlying harmonic structure:

A minor→D Major→E minor→F Major→E Major→A minor→F Major→C

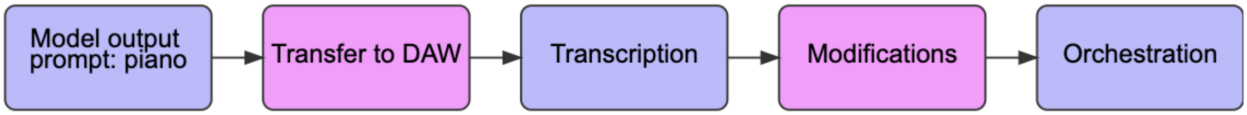


Figure 5.6: The workflow of cue 3 of *Infinite*

Major→A minor→Bb Major

Using the transcribed material as a foundation, I orchestrated the piece with strings, pads, and wind instruments, aiming to retain the core ideas of the model's output while mitigating its overly sentimental tone. The melody was reworked to allow for more phrasing and dynamic breathing, and it was distributed among different instruments with doublings to introduce textural variety and new timbral colors. Although the original output served as a reference for tempo and meter changes, the orchestration itself was entirely reimagined.

Despite these efforts, the final product retains a certain "heavy-handedness" for the specific scene in question. Nevertheless, this example demonstrates how generative AI can assist composers by providing a foundation or reference point for creative work. While the output required significant refinement, it highlights the potential for AI tools to augment the compositional process, particularly in generating initial material that can be shaped into a final product through human intervention.

Before actually scoring with the assistance of this model, I had hoped that it would behave more like an assistant, where I would be able to generate musical ideas coherent to my harmonic and instrumental styles to inspire me in the scoring process, or that it could act as an orchestrator, where I would give it a piece of melody as a conditioning element for it to orchestrate in a particular style. However, upon testing this method, the experience feels backward. The model acts more like a boss who would give me a general idea and then expect me to orchestrate and conform it. Moreover, this process takes longer than simply writing everything myself, for the generated content would often have low fidelity and, therefore, be unusable in a cue. The low resolution can be remedied to an extent by enabling multiband diffusion, where the model would infer at a higher sample rate. However, if realism were to be considered a criterion, the current model that is accessible to everyday composers would not suffice. The generated outputs would often sound like someone who had heard much music and has some ideas of the timbre of different instruments; they proceed to create what they had heard in their fever dream despite the pre-trained model being trained on a large corpus of live recordings. If a larger model running on something with more computing power were to try the same thing, the result would be significantly better. However, it would not be accessible to composers every day, especially without having them upload their works to a server somewhere. The output audio files cannot be compared with mockups due to the model's uncontrollable nature; the results often sound like a mishmash of realistic instruments being vaguely remembered and playing at a low resolution. There are no ways (at least not with this architecture) to control the parameters such as BPM, key, chord changes, etc. Therefore, there is no way to enhance realistic programming with this particular model at the moment.

5.4 Where does VR fit into all of this?

The research and experiments conducted in this study are intended to serve as an initial step of a more extensive and ambitious project. During the composition of my dissertation, I also had to write the score for a new VR game in addition to rescoring the cues from two already released films. Traditional films, being in the 2D format, give the composers limited freedom in expressing their musical ideas. However, that freedom increases exponentially with VR, especially interactive VR media and all other forms of modern media; the experiencers are no longer confined within fixed camera angles and are encouraged to explore their surroundings. Having scored a VR game, I now understand the importance of music and sound in the context of total immersion. An interactive musical element is crucial to creating an immersive environment for the person experiencing the VR environment.

Though there are ways to achieve this effect by using techniques such as triggers, pre-composed layering remains the dominant method for creating reactive and immersive soundscapes. This approach involves preparing multiple musical layers or cues that can be triggered based on specific player actions or environmental changes. When I composed the score for the VR game *Land of The Forgotten*, I was first given a demo of the game environment. My collaborator and I then discussed the layout of the game's stages, identifying the general atmosphere of each area and determining the locations for triggers. We agreed to produce multiple tracks of the same length to maintain cohesion. For "Chapter One," I first composed a track featuring ambient pads and a rocking bass line,[5] which I called the "base layer." This layer was designed to loop continuously throughout the stage. Subsequently, I created five additional layers, each triggered by either the player's location or gaze. These layers build upon the base track, with harmonization tailored to reflect the unique characteristics of the different areas within the stage.

For "Chapter Two," we employed a similar approach, dividing the stage into two main areas, each further subdivided into three smaller sections. As in Chapter One, I composed tracks of the same length designed to loop throughout the stage. However, instead of combining distinct layers with the base layer, this stage adopted a progressive structure. Layers were incrementally added on top of one another, creating an evolving musical texture that corresponded to the player's advancement through the stage. Upon reaching the final temple area, the layers continued to build, culminating in a moment where all elements were suddenly stripped away, leaving only a solitary piano track. This final piece conveyed a subtle sense of melancholy, encouraging the player to reflect on the journey they had just

[5] A pattern characterized by alternating or repeated notes that create a driving, rhythmic foundation

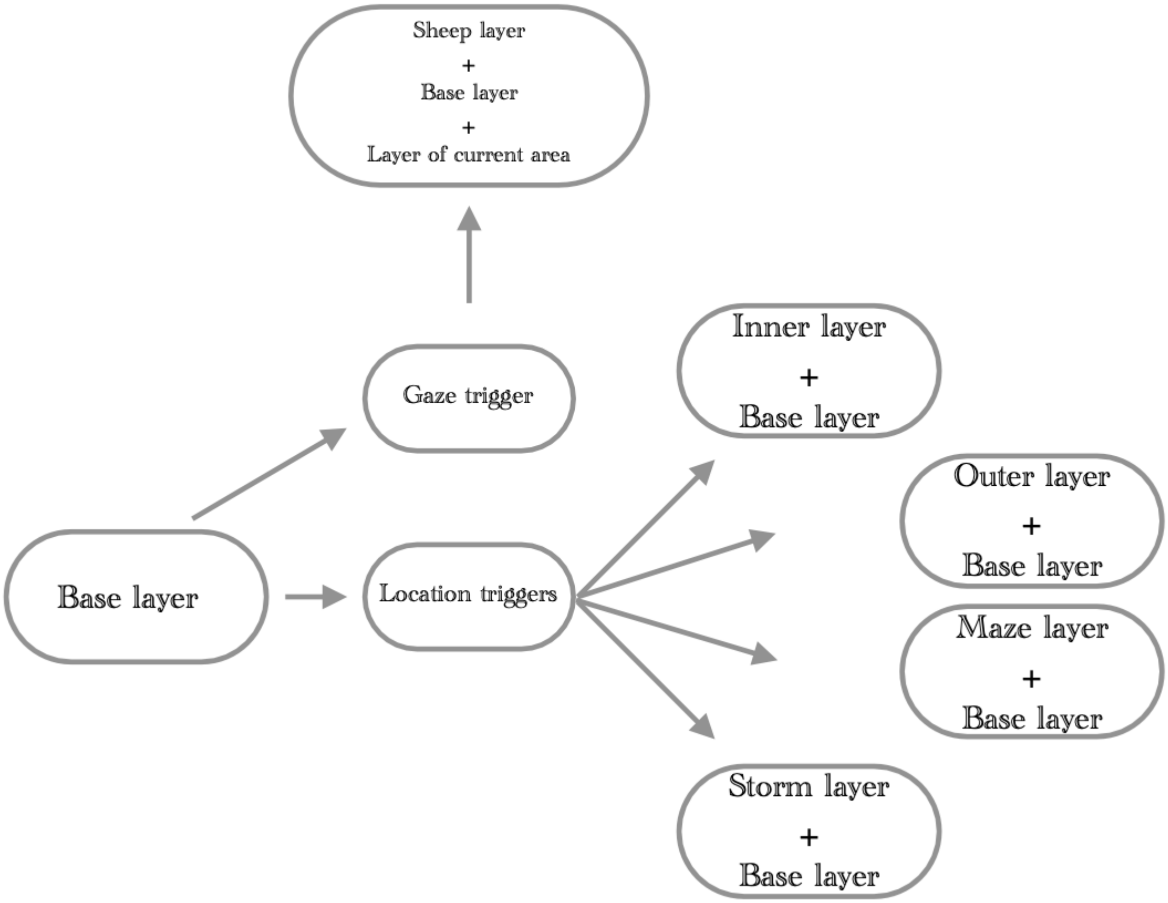


Figure 5.7: Triggers and layers in the first chapter of *Land of the Forgotten*

experienced.

The elegance of this method lies in its reliability, precision, and the level of control it grants composers over the aesthetic outcome. For example, a sudden shift in a game's narrative might seamlessly transition between pre-written musical layers to heighten emotional impact. However, this approach is inherently static and finite, constrained by the pre-existing assets and the specific scenarios envisioned by the composer.

Generative dynamic music systems, on the other hand, represent an attempt to move beyond these limitations by introducing algorithms or models that can compose music in real

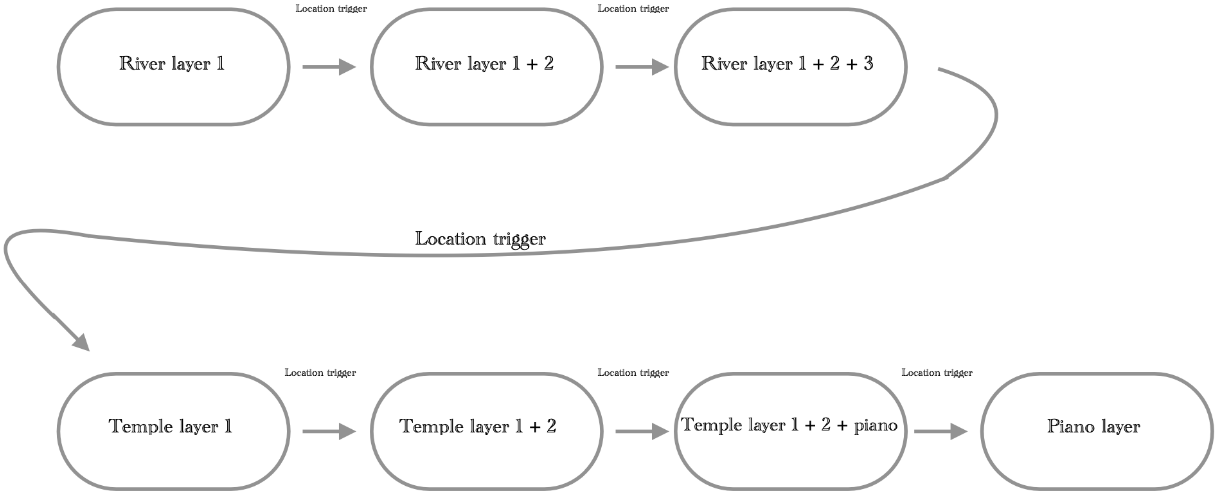


Figure 5.8: Triggers and layers in the second chapter of *Land of the Forgotten*

time based on the player's actions or the environment. While the potential of such systems is clear, their current implementations often fall short of the polish and intentionality that pre-composed layering achieves. Generative systems, particularly those driven by AI, can sometimes produce musical outputs that feel disconnected, lack thematic coherence, or fail to align with the emotional arc of the experience. These shortcomings highlight the difficulty of balancing the computational possibilities of real-time music generation with the artistic sensibilities of a carefully composed score.

Despite these challenges, I believe generative systems hold significant promise for the future of VR. Unlike pre-composed methods, generative music can adapt to unanticipated player behaviors or emergent gameplay scenarios, creating a truly dynamic and personalized auditory experience. The ultimate goal for this project is to be able to have a model that not only can generate music in the composer's particular musical style but does it in real-time in response to what is happening within the experiencer's field of view. Traditionally, this is done by the axis information gathered from the headset's accelerometer. [6] However, the combinations are limited and, therefore, cannot achieve a unique experience tailored to a specific experiencer. Hypothetically, an AI-driven system could analyze player actions, spatial positioning, and even biometric feedback (such as heart rate or gaze direction) to generate music that evolves in real time. This capability could enable the music to feel more alive and integrated into the VR environment, blurring the line between reactive sound design and interactive composition.

The path forward likely lies in hybrid systems that combine the strengths of both approaches. By layering generative elements on top of pre-composed frameworks, composers could retain the thematic consistency of traditional methods while leveraging the adaptability of AI-driven systems. This fusion could allow VR experiences to achieve a new level of musical interactivity and immersion without sacrificing artistic integrity.

A fine-tuned model could be combined with a computer vision program such as YOLO (You Only Look Once.)[7] It is a fast and efficient object detection system that processes an entire image in one pass through a neural network. It divides the image into a grid, and each part of the grid predicts whether it contains an object, where the object is, and what type of object it might be. For each detected object, YOLO predicts the object's position (center, width, and height) and how confident it is about the detection. During testing, it combines this information to determine the probability of each object and how well the predicted box matches the object.(28)

YOLO is quick enough to work in real time, making it useful for tasks like detecting objects in video streams. Unlike older methods, which might focus on parts of the im-

[6] A sensor that measures the speed and direction of your head's movements, helping to track motion

[7] A real-time object detection algorithm in computer vision that processes an image in a single neural network pass to identify and locate objects

age separately, YOLO considers the entire image at once, helping it avoid false detections and better understand the context. For example, it is less likely to confuse a shadow in the background for an object because it processes the whole scene (28). Hypothetically it could recognize objects and then converts that information into text prompts for the music generation model to generate new music that perfectly aligns with the objects on the screen.

While generative systems are not yet as refined or elegant as pre-composed layering, they offer a glimpse into a future where music is not merely reactive but symbiotic with the player's journey. The challenge lies in bridging the gap between technical innovation and compositional artistry—a challenge that I find both daunting and exciting as I continue to explore the potential of these tools in my work.

CHAPTER 6

Some very necessary discussions

The integration of AI into music composition has sparked a significant debate within the academic and creative communities. Because this technology is still in its infancy, lawmakers are still in the process of carving out the ethical and legal frameworks to address the issues that may or may not emerge from its use. My position is impartial, as I am interested in observing where it progresses. While I acknowledge concerns about how it is currently being handled, I do not believe that dislike of its present state is a sufficient reason to reject it entirely. Moreover, I do not believe that we, as creators, should or can be replaced by it. Proponents argue that generative AI serves as a powerful tool that enhances creativity, democratizes music production, and enables novel forms of artistic expression. For instance, studies by Briot et al(29) demonstrate how AI can assist composers in exploring complex musical structures and generating innovative melodies that might be unattainable through traditional methods. Additionally, AI-driven platforms like Meta's MusicGen and Google's MusicLM have lowered barriers to entry, allowing individuals without formal musical training to create sophisticated compositions.

Conversely, critics raise ethical and artistic concerns regarding AI-generated music. A primary objection is the potential devaluation of human creativity and the risk of homogenization in musical styles, as AI systems often rely on existing datasets that may perpetuate prevalent trends (11). Additionally, some argue that AI-generated compositions should not be classified as true music because they ostensibly lack the composer's personal emotion, musical depth, and life experience. However, this viewpoint can be challenged on several grounds. First, a composer's emotional input and musical depth can be integrated into AI models through the expression of emotions and detailed descriptions of musical intricacies provided as text prompts. Second, authentic life experience has never been an absolute prerequisite for composing music. Historically, composers and songwriters have effectively conveyed experiences without personally undergoing them. For instance, Irving Berlin, a Russian-born Jewish composer who never celebrated Christmas, wrote "White Christmas," one of the most iconic American holiday songs. This example illustrates that the absence of direct personal experience does not necessarily impede the creation of emotionally resonant and meaningful music.

Through this research, we have demonstrated that individual composers can effectively harness generative AI models to create unique musical compositions by integrating their own musical ideas through text prompts and fine-tuning techniques. This methodology enables composers to input specific creative directives and adjust AI-generated outputs to reflect their personal artistic vision. By doing so, the collaboration between AI and human composers hypothetically mitigates the risk of stylistic homogenization. Furthermore, this partnership preserves and enhances the value of human creativity, as composers retain control over the emotional and structural elements of their work. Our findings illustrate that generative AI can serve as a complementary tool, facilitating the exploration of novel musical landscapes while maintaining the distinctiveness of each composer's individual style. This synergy not only fosters innovation in music composition but also ensures the continued diversity and authenticity of artistic expression. Consequently, this study underscores the potential for AI-human collaborations to advance the creative process without compromising the unique contributions of human artists. To further emphasize the ability of individual composers to achieve this, we intentionally selected the MusicGen model, which is open-sourced and can be run and fine-tuned locally on a consumer-grade laptop.

Experiments like this open the possibility of composers adapting to this new age of generative AI, increasing the competitiveness of individual composers by boosting their productivity, making their music production speed somewhat on paar with big techs' commercial models, especially when experts have predicted an explosion in contents(30), be it videos, arts, or music, in the next few years. This study is designed to provide a solid, replicable example demonstrating that AI can function potentially as an assistant. While it represents only an initial step, it offers a foundation upon which future work can build. While this work aims to help composers protect and maintain their livelihoods, one potential issue that stems from this accessibility is the possibility of a composer feeding their model with someone else's works. In this case, there will need to be measures, such as antipiracy codes embedded in the codec of audio files of other composers' works, ensuring that it would be impossible to use their works to train one's model without their consent.

Another contentious debate arises regarding authorship and intellectual property rights in the context of AI-generated works. It questions who holds ownership over such creations—the algorithm developers, the composers whose music is used for inference, or the end-users who generate the music. Furthermore, there is concern about the potential displacement of professional musicians and composers, as AI tools may diminish the demand for human expertise in the creative process(10).

For this work, we touch on this controversy by trying to maximize individual composer control and ownership by using MIT-license open-sourced model, which allows the user to use the model for private or commercial use and free to modify and distribute. While the model weights are licensed under the Creative Commons Attribution-Non Commercial 4.0 International (CC BY-NC 4.0) license. At the time of this work, it is unclear whether the output generated by the model is automatically subject to the same licensing restrictions as the model weights,[1] since the output is not directly part of the weights.

This ongoing controversy underscores the paramount importance of sustained discourse and investigation into the role of AI within creative industries. It is imperative to comprehend the delicate balance between technological advancement and the preservation of human artistic value, as this understanding is crucial for formulating policies and practices that promote innovation while upholding ethical standards. Consequently, this work contributes to a novel approach of how composers can utilize generative AI in the realm of music as a collaborative tool and remain in control throughout the entire creative process.

On the other hand, what does it mean when a composer successfully trains a model that is fine-tuned to their musical styles? After all, if one simply trains a model and then stops writing, the musical brain of that composer stops evolving, meaning the model would stay the same and keep writing the same music. How does one teach the model the unknowable? Therefore, having a model can only mean one thing – that composer now has access to a free assistant that writes what they tell it to write; it will not write new music unless they teach it to the model. Having trained a model does not mean one can stop writing music; on the contrary, one needs to keep composing, growing, and feeding the new music to the model so the model can evolve with the composer. It would be unfathomable if I still wrote the same music I used to write when I was thirty years old at sixty years old.

I would like to emphasize that the technology itself is not the adversary in this context; rather, the challenge lies with individuals or entities that misuse it and exploit the works of others. As lawmakers continue to develop ethical and legal frameworks to address the potential issues arising from AI technologies, it is crucial for creatives to understand the foundational principles of how these systems operate. By doing so, we can actively partic-

[1] The adjustable numbers in the model that determine how it processes input data and generates output, based on what it learned during training

ipate in the necessary discussions and ensure that our voices are heard and our concerns represented.

Without this understanding, there is a risk that our perspectives may be dismissed or deemed uninformed. Blindly boycotting the technology without engaging with its mechanisms and implications could leave us marginalized in critical conversations, allowing others to control the narrative and policies that directly impact us. By equipping ourselves with knowledge and presenting a unified, informed voice, we can advocate effectively for ethical practices and safeguard the integrity of our creative industries in the evolving technological landscape.

CHAPTER 7

Conclusion

This dissertation has explored the potential of AI-generated music as a tool for modern composers, focusing on the development of models that prioritize accessibility, adaptability, and ethical considerations. By examining the historical trajectory of AI in music, from early algorithmic compositions to advanced neural networks, it becomes clear that while these tools have the capacity to augment artistic processes, they are fundamentally contingent on their training data and the guidance provided by human creators.

Central to this work was the pursuit of an open-source AI model capable of running locally and being fine-tuned to emulate the stylistic nuances of specific composers. This objective reflects a broader goal: to empower composers with tools that enhance their creative agency while protecting their intellectual property from the pervasive practice of data scraping. The ethical challenges posed by generative AI technologies, particularly in the context of corporate control and the commodification of artistic labor, underscore the urgency of reclaiming creative autonomy in the age of AI.

The fine-tuning and evaluation of MusicGen demonstrated both the promise and limitations of current AI systems. While these models can effectively learn and replicate stylistic elements, their outputs are constrained by the data they are fed and lack the intuitive, experiential understanding that defines human creativity. Despite these limitations, AI could offer immense potential as a collaborative partner, capable of handling repetitive or labor-intensive tasks, thus freeing composers to focus on higher-level creative decisions.

Throughout this dissertation, I have expressed optimism about the potential of AI to democratize creative tools and empower independent composers. By making sophisticated generative models more accessible, AI could theoretically level the playing field, providing smaller creators with resources that have traditionally been out of reach. However, I acknowledge the complexity of this vision and the skepticism it invites, particularly when considering the entrenched power of larger entities in the industry.

When referring to larger entities, I am speaking broadly of two intersecting spheres: Big Tech companies, which own and control the most advanced AI tools and infrastructure, and composer collectives like Remote Control Productions or Bleeding Fingers, which dominate high-profile scoring projects through economies of scale and deep industry ties. Both entities represent significant barriers to entry for independent creators. Big Tech's proprietary models are often too costly or restrictive (not to mention the fact that they could be unethical,) for smaller creators to access fully, while established composer collectives maintain industry monopolies on blockbuster projects, leveraging their reputation and resources to outcompete newcomers.

Despite these challenges, I believe there is room for optimism, albeit a cautious one. open-source initiatives and smaller-scale AI tools are beginning to emerge, providing pathways for independent creators to experiment and develop within their own artistic spaces. While such tools may not yet rival the capabilities of proprietary systems, they are a step toward empowering individuals to integrate AI into their workflows without complete reliance on corporate-controlled platforms.

I recognize, however, that the current dynamics still favor those with access to substantial resources—whether through corporate alliances, funding, or industry connections. My optimism is not a dismissal of these realities but rather a reflection of the potential for future progress. AI's impact on the creative landscape will ultimately depend on how these tools are distributed, governed, and utilized. To truly equalize opportunities, the broader creative and technological communities must address issues of accessibility, education, and ownership.

This research represents the first step in a larger endeavor. Future work will aim to develop AI systems that not only emulate specific styles but also generate real-time musical responses tailored to immersive environments. Such advancements could transform the way music interacts with dynamic media, allowing for personalized and adaptive scoring in virtual and augmented realities. By bridging the gap between technological innovation and artistic integrity, this project aspires to ensure that composers remain at the forefront of creative expression in a rapidly evolving digital landscape.

Ultimately, this dissertation advocates for a vision of AI that serves as a tool of empowerment rather than one of exploitation. I aim to contribute to this ongoing dialogue by exploring the ways in which AI tools can support creative independence, even within a challenging landscape. While the barriers are real, the promise of these technologies lies in their potential ability to inspire and enable creativity in ways that transcend current limitations—if we are deliberate about how they are developed and shared. By equipping individual creators with accessible, transparent, and customizable technologies, we can foster a future where the intersection of AI and music enriches human artistry while safeguarding the rights and agency of those who make it possible. In doing so, we reclaim the creative domain from the control of large corporations and preserve the deeply human essence of music in the face of technological change.

Bibliography

[1] A. Roberts, J. Engel, Y. Mann, J. Gillick, C. Kayacik, S. Nørly, M. Dinculescu, C. Radebaugh, C. Hawthorne, D. Eck, Magenta studio: Augmenting creativity with deep learning in ableton live, academic.edu (2019).

[2] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, I. Sutskever, Jukebox: A generative model for music, arXiv preprint arXiv:2005.00341 (2020).

[3] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, Y.-H. Yang, Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[4] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, et al., Musiclm: Generating music from text, arXiv preprint arXiv:2301.11325 (2023).

[5] A. Défossez, J. Copet, G. Synnaeve, Y. Adi, High fidelity neural audio compression, arXiv preprint arXiv:2210.13438 (2022).

[6] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, A. Défossez, Simple and controllable music generation, Advances in Neural Information Processing Systems 36 (2024).

[7] I. E. Sutherland, A head-mounted three dimensional display, Proceedings of the December 9-11, 1968, fall joint computer conference, part I (1968) 757–764.

[8] L. A. Hiller, Computer music, Scientific American 201 (6) (1959) 109–121.

[9] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, D. Eck, A hierarchical latent vector

model for learning long-term structure in music, in: International conference on machine learning, PMLR, 2018, pp. 4364–4373.

[10] F. Morreale, Where does the buck stop? ethical and political issues with ai in music creation, archive.org (2021).

[11] V. D. Kirova, C. Ku, J. Laracy, T. Marlowe, The ethics of artificial intelligence in the era of generative ai, Journal of Systemics, Cybernetics and Informatics 21 (4) (2023) 42–50.

[12] K. Glaskin, Dreams, memory, and the ancestors: creativity, culture, and the science of sleep, Journal of the royal anthropological institute 17 (1) (2011) 44–62.

[13] R. Cartwright, The twenty-four hour mind: The role of sleep and dreaming in our emotional lives.

[14] T. D. Dillehay, J. Rossen, D. Ugent, A. Karathanasis, V. Vásquez, P. J. Netherly, Early holocene coca chewing in northern peru, Antiquity 84 (326) (2010) 939–953.

[15] G. Y. Kostov, Fostering player collaboration within a multimodal co-located game, FachhochSchule, Hagenberg (2015).

[16] L. Snapes, 'he touched a nerve': how the first piece of ai music was born in 1956 (dec 2021).
URL https://www.theguardian.com/music/2021/dec/07/he-touched-a-nerve-how-the-first-piece-of-ai-music-was-born-in-1956

[17] C. Roads, Artificial intelligence and music, Computer Music Journal 4 (2) (1980) 13–25.

[18] A. Buchner, I. U. Lewitová, Mechanical musical instruments, (No Title) (1959).

[19] H. A. Simon, R. K. Sumner, et al., Pattern in music, Formal representation of human judgment. New York: Wiley (1968) 219–250.

[20] D. Cope, Experiments in musical intelligence (emi): Non-linear linguistic-based composition, Journal of New Music Research 18 (1-2) (1989) 117–139.

[21] D. Cope, Virtual music: computer synthesis of musical style, MIT press, 2004.

[22] O. Bown, Experiments in modular design for the creative composition of live algorithms, Computer Music Journal 35 (3) (2011) 73–85.

[23] M. Tracy, Major record labels sue a.i. music generators (Jun 2024).
URL https://www.nytimes.com/2024/06/25/arts/music/record-labels-ai-lawsuit-sony-universal-warner.html

[24] H. Bray, Record companies accuse ai music startup suno of "copyright infringement on an almost unimaginable scale" - the boston globe (Jun 2024).
URL https://www.bostonglobe.com/2024/06/24/business/suno-ai-music-lawsuit/

[25] S. B. Griffith, Sun Tzu: The art of war, Vol. 39, Oxford University Press London, 1963.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 30 (2017) 5998–6008.

[27] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, L. Zhao, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, B. Ge, Summary of chatgpt-related research and perspective towards the future of large language models, Meta-Radiology 1 (2) (2023) 100017. `doi:10.1016/j.metrad.2023.100017`.
URL http://dx.doi.org/10.1016/j.metrad.2023.100017

[28] J. Redmon, S. K. Divvala, R. B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, CoRR abs/1506.02640 (2015). arXiv:1506.02640.
URL http://arxiv.org/abs/1506.02640

[29] J.-P. Briot, G. Hadjeres, F.-D. Pachet, Deep learning techniques for music generation, Vol. 1, Springer, 2020.

[30] D. Bazaraa, Experts predict "explosion" of deep fakes used in tv and film with 90% of online entertainment content being ai generated by 2025 – as critics and viewers pan itv comedy that replicates greta thunberg, stormzy and harry kane (feb 2023).
URL https://www.dailymail.co.uk/news/article-11700593/Experts-predict-explosion-deep-fakes-90-online-content-AI-generated-2025.html?ito=email_share_article-top