UTRECHT UNIVERSITY

Department of Information and Computing Science

Proposal - Master Thesis - Artificial Intelligence

Rhythm and Reason: Adding fine-grained control to deep learning music generation models.

First examiner:
Anja Volk

Candidate:
Efraim Dahl

Second examiner:
Peter van Kranenburg

February 3, 2025

Abstract

Music is essential in video games, enhancing immersion and engagement, but players may disengage due to excessive repetition or personal preferences. This is particularly problematic in games like LastMinuteGig, a Musical Attention Control Training (MACT) application for Parkinson's patients. To maintain engagement and the effect of the intervention, a diverse set of controlled, adaptive music is needed. Advances in generative music offer a scalable solution. This research explores efficient methods of adding control to pre-trained models, to steer towards rhythmically adaptive music suitable for MACT. Insights from small-scale experiments will inform the application of rhythmic control in MusicLang, a transformer based music generator, which will be used to build musical assets for LastMinuteGig.

Contents

1. Introduction

Music is a cornerstone in modern video games. Music evokes emotional responses, triggers memories, can direct attention and improve immersion and the overall experience of a game. Yet, it is not uncommon for players to become disengaged from the game music, often due to excessive repetition or simply different personal preferences.[1] This can be a problem in "serious" games for therapeutic use such as "Last Minute Gig" a gamified Musical Attention Control Training (MACT) application, aimed to improve attention control in patients with Parkinson's disease [2]. Ideally MACT is a personalized experience with dynamic adjustments taking into account a patient's abilities and preferences. In the context of gamified self-directed therapy,this is crucial to keep patients engaged and progressing.[1] In order to accommodate a wide array of patient's abilities and preferences in a MACT-game, a considerable amount of musical, specifically musical material that fits the constraints of MACT is required. One potential avenue to provide this is through controlled music generation.

Over the last years, music generation has grown from a mostly academic endeavor to a billion-dollar industry. Large technology companies are exploring music generation with foundation models such as Meta's MusicGen[3] or Google's MusicLM [4]. The commercial startup Suno is valued at 500 million dollars, while chart-topping songs are being generated with Udio [5][6]. This breakthrough is powered mostly by advances in language modelling, bringing about Large Language Models such as GPT-4 or LLAMA3 applied to music.

In the context of games, generative music is promising to cater to different user preferences and enable a richer, more varied musical experience. Video games can include hundreds of hours of game-play and branching story lines. When additional complexity is added to adapt the music to individual preferences, it quickly is no longer feasible to manually compose unique music to fit all scenarios. Especially in more resource constrained applications such as games for therapeutic use, generated music can offer an avenue to provide a large variety of music that can improve user engagement, and ultimately the effect of the intervention. One of the key components of a successful generative model is control. In music generation control could be a text-description, information about genre, style or instrumentation. Control can also include fine-grained musical parameters such as a melody, chord progression or musical structure. Generative models can often be controlled with human-composed music,

[1] This type of therapy is not aimed at replacing traditional guided therapy. Ideally it is used as supplement, and may provide relief in situations where traditional therapy is not immediatly available

where the provided music is continued, interpolated between or inpainted (generating accompaniments, melodies, or additional musical lines). In LastMinuteGig[2] the music periodically provides stimuli, a noticeable change in the music, based on which the user changes their playing, specifically their rhythm. One way this stimulus could be achieved in generated music is by adding control that introduces a shift in rhythmic patterns at semi periodic intervals.

There are a variety of ways to achieve control in a music generator. For rule based systems, control is added explicitly, in statistical systems (including deep learning) control is typically applied through architectural constraints and choice of training data. Certain deep learning architectures lend themselves to certain types of control. I.e an auto-regressive transformer models a sequence based on prior parts of the sequence, it natively generates continuations of an input. In deep learning, other musical parameters (i.e chord progression or melody) can be controlled for by joint conditioning of a model, which means that the controlled musical parameters are made explicit while training. This method of adding control is quite cost-intensive requiring as much, or even more training material as the baseline model without controls.

A more economical method of adding control to a deep learning generative model is through fine-tuning and post-hoc conditioning. In fine-tuning (sometimes referred to as transfer learning) a pre-trained model is trained on additional data. There are various methods of fine-tuning, involving different configurations of the model, the model parameter and the integration of control input, but typically fine-tuning uses considerably fewer resources and data than training a model from scratch. In post-hoc conditioning a generative model is adjusted while it is running to increase the likelihood of the output to fit the desired constraints. The goal of my thesis is to generate a large amount of music to extend the LastMinuteGig video game for MACT in patients with Parkinson's. LastMinutGig is a mobile application built with Unity3D, in its current iteration it contains a single button with which a player taps along to the music, controlling a set of guitar chords. When a musical change occurs the player is instructed to change their tapping. For this I will first explore and evaluate different methods of adding control to a deep learning model on a small generative music model. I will apply the most successful techniques of adding control specifically of rhythmic patterns learned from this first step to a larger generative music model, MusicLang to steer the output towards something usable in the MACT context.

2. Literature

The following section contains an extensive (though not exhaustive) categorical literature study. The first part provides contextual information, including an overview of the history of music generation, motivations and ethical concerns. The second half is more technical with a discussion on basic concepts such as music representation and recurring challenges in music generation such as control. This is followed by a high-level discussion on the techniques used in music generation in the past. Finally, I will be discussing in more detail the models most used in recent research. In particular I will review different methods of adding control for these models. As a starting point I used the most recent ISMIR (International Society for Music Information Retrieval) papers on music generation and their referred sources alongside other papers suggested by my advisor Anja Volk. In addition I performed systematic searches with the following keywords. Deep Learning Music Generation, Diffusion Music Generation, Transformer Music Generation, Symbolic Diffusion Music Generation, Control Music Generation.

2.1 Brief History of Music Generation

From antique wind-chimes to classical period musical dice games, the aleatoric music of the 20th century, all the way to deep learning based music generators -humans have used algorithmic, probabilistic, and statistical methods to create music. As early as the late 1940s computers have played a role in composition as sound generators and instruments [7] and providing musical material themselves such as in the 1957 Illiac suite [8]. Over the next 50 years, there are a lot of mostly academic, quite disparate experiments in music generation most of them in the symbolic music domain. They utilize a variety of contemporary AI technologies, from expert systems and ontologies to evolutionary algorithms to feed-forward, recursive, and convolutional neural networks. Some composers in the classical tradition such as Iannis Xenakis [9] and David Cope [10] use computer algorithms in their creative work. In the 2010s a small ecosystem of commercial generative music startups such as Jukedeck, PopGun, and Ampermusic [11] started to emerge, alongside an increasing number of publications applying deep learning, particularly GANs, and RNNs to music including MIDInet [12], [13] and Sturm and Ben-Tal's work on FolkRNN starting in 2016 [14]. With the development of the transformer architecture in 2017 (Vaswani et al., 2017), large technology companies start experimenting with music generators including OpenAI's Jukebox [15] and Musenet[16], Meta's MusicGen[3] and Google's MusicLM [4] and the preceding Magenta project, with fully generative models ca-

pable of producing sequences of high-quality music modeled from raw audio. At the time of writing, commercial music generators such as Suno and Udio are raising millions of investment funds[6], while generated music is being widely streamed and actively used in TV and video productions.

2.2 Why generate music? - Motivation

2.2.1 Composition co-pilots

The earliest music generators (including the 18th century musical dice games), where justified as methods to deliver inspiration to composers and music makers, including enabling novice composers to write music. Composer David Cope[10], states that he turned to music generation to overcome writer's block. Commercial enterprises like Suno, claim to be "building a future where everybody can make great music" [17] Many current and past efforts point to music generation as a process that accompanies composers. In DeepBach[13], the authors go through considerable difficulties in building the system to be very flexible and usable in real-life composition systems. Initially, they developed a muse-score integration and later tied DeepBach into the web app NONOTO[1] to support musical inpainting in Ableton Live scores. Similarly with Composer's Assistant[18] the authors explicitly enable musical inpainting and continuation in the REAPER music production software. These efforts to make music generators accessible to a wider audience of music makers by integrating them into common software are commendable and are an important step for musical co-creation.

2.2.2 Music in games

Beyond the co-creation of music, there is a range of music that serves a functional rather than a purely aesthetic purpose. This can be background music in videos and therapy assisisting games. In games, one can make the case for AI-generated music or AI-assisted music due to their size and interactability. Video games can facilitate often hundreds of hours of gameplay, they can have branching storylines and complex player interactions. Game sound tracks typically dont cover more than a fraction of that time.[19][20]. Through different adaptive techniques, relatively short snippets of the original material can be stretched into hours of unique audio often relying on the recombination of different elements, but rule-based recombination only goes so far. A recent study of player behavior [1] finds that many players turn off the game music eventually. They cite different reasons, i.e. preference for their own music over the game music (46.7%) or the repetitiveness of in-game music (29.6%).

[1] https://github.com/SonyCSLParis/music-inpainting-ts

The procedural generation of 3D assets and levels and enemy behavior is quite commonplace. This is rarely applied to music. Composing and adapting music to fit every scenario possible would be tedious. With AI-assisted music composition, it could become possible to enable adaptive audio on a large scale. Either by creating a large number of musical assets anticipating player choice, or generating new variations of the game score on the fly, improving player immersion through real-time adaptation. Several challenges make music generation difficult in games. There are performance issues: AI generators are often resource intensive. Additionally they are difficult to control, and there is little guarantee that the generated tracks will be appropriate. [19] Currently, they also lack proper integration into video game environments and engines. [20].

2.2.3 Music in serious game

Aside from games for general audiences, there is a potential use for generative music within serious games in the context of music therapy. In music therapy, music can be used for emotion regulation, motivation, adherence, motor coordination, (rhythmic) entrainment, and to facilitate social interactions. [21] Musical attention control training for instance has been shown effective in helping patients with Parkinson's[22], ADHD [23], autism [24] and psychosis [25] improve their mental capabilities for selective attention and switching attention. Serious games have the potential to supplement music therapy. "Last Minute Gig" [2] implements clinical music therapy protocols as a serious game to improve attention control in Parkinson's patients. This thesis works towards expanding the game with generated music to improve player experience.

2.3 Why not generate music - Ethical Concerns

2.3.1 Introduction

There are several concerns related to the AI-based generation of music. First, there are basic legal concerns about copyright and licensing. During the generation process generative models may produce output that is identical or very similar to (copyrighted) training data. More fundamentally, during training the question remains whether models should be allowed to train on copyrighted data in the first place. More broadly, and this also applies to non-deep learning-based generators, are concerns about the devaluation of human labor and creativity, the flooding of our cultural spaces with low-quality generated content. Finally, there are concerns about the environmental impact, especially of large generative models, which require substantial amounts of energy, water, and rare materials to run.

2.3.2 Data Leakage and Copyright

Generative AI companies are achieving record-breaking valuations and this includes music generators with Suno at a valuation of about 500 million dollars after a 125 million dollar fundraiser leading the pack. [6] [26]. However, AI companies are facing backlash from artists and record labels with an organization of record labels including the "big three" Sony, Warner Music, and UMG suing Suno and Udio for $150.000 dollars per infringed work[27]. Generative AI runs a substantial risk of parroting or leaking training data. In language models, the leakage problem is of concern when training on data that contains sensitive information, which may be revealed either through accidental leakage or through membership inference attacks [28]. While leakage may raise privacy concerns in other generative models such as speech and image-generators [29] for music, the risk of training data leakage is mostly an issue of copyright. Ed Newton Rex shows some examples of how Suno can be influenced [30] to leak training data. This ability to create disconcertingly close reproduction of copyrighted work is also cited in the court documents. Suno has since started to prevent prompting with artist names (i.e. in the style of Eminem) and including known song lyrics.

2.3.3 Training and Copyright

Besides leaking training data, there is the more general question of whether AI models should be allowed to train on unlicensed work. Echoing the court case between OpenAI and the New York Times [31], both Suno and Udio cite fair use in response to accusations of copyright infringement. In US copyright law the fair-use clause limits exclusive rights to a work, with four factors to consider: 1) purpose and character of the work in use, 2) nature of copyrighted work, 3) amount of the copyrighted work used, and 4) the effect on the potential market or value of copyrighted work.[32] Fair use is often granted to derivative works such as parodies and covers and works used in educational settings. AI's learning of structures has also been likened to the human learning process, humans learn based on copyrighted music they listen to, without giving credit or compensation to their influences. Newton-Rex [30] rejects this comparison. In learning music, humans contribute to the musical ecosystem, they take lessons, go to performances, or at the very least generate some streaming revenue for artists, none of these are true for machine learning models learning from scraped data.

2.3.4 Devaluing Music

While few are following director Ram Gopal Varma's announcement to only use AI-generated music in his future films [33], AI-generated music is becoming increasingly difficult to differentiate from human production and already receiving considerable amounts of streams and it is not unlikely that music in film, video and game projects may be replaced or at least sup-

plemented with AI-generated music. The online music market is saturated, with more than 100,000 songs uploaded to music streaming giant Spotify every day [34]. Generative AI may just further exacerbate this problem, resulting in a race to the bottom for creatives and musicians.

2.3.5 Environmental Impact

Digital infrastructure, traditional data centers, crypto-currency mining, and AI-centered data infrastructure account for about 2% of the world's energy consumption [35]. The performance of current large language models often scales with simultaneous increases in model size, training data, and computation time.[36] Each of these three factors requires considerable resources. Music generation is no exception. In a recent ISMIR publication [37] the authors make estimates on energy consumption of different projects related to music generation and computation-intensive MIR, finding an average energy consumption of 224.8kWh for model training (the energy consumption of an average western person over 2 months). The energy consumption is divided highly unevenly, with the median being at merely 18 kwh (3 days of an average westerner's energy consumption). Music generation models associated with large technology companies are responsible for about 89% of the estimated energy use. This is only for training, models that are deployed publicly, continue to use substantial energy for inference. Beyond just the carbon footprint of generative AI, the local impacts of resource use such as rare minerals and water are important to keep in mind.

2.3.6 How I plan to address these concerns

During this thesis process, I'm planning to address the outlined ethical concerns in the following manner. First, I will only train, and use open-source models trained on licensed data providing attributions where possible. I will also make my trained models publicly available with substantial documentation to contribute to the open-source ecosystem, research, and music. Second I am explicitly designing the generative process to be cooperative, this is facilitated on the one hand through the introduction of new control modalities, and on the other the choice of symbolic music over audio, which eases editing and integration into composition software. Finally, I am not attempting to train custom large foundation models, rather I'm trying to find ways to extend existing models to introduce new control mechanisms, requiring training on only a fraction of the model parameters, with substantially less need for data, computation, and energy.

2.4 Representation

2.4.1 Symbolic vs Audio

Music can be digitally represented in two ways, either directly as audio or symbolically as a digital score. Working with different representations comes with different drawbacks and benefits. Symbolic data is inherently a condensed form, it contains no or very little information on many acoustic features such as timbre and acoustic space, it is also less precise in terms of timing and pitch due to inherent quantization when transcribing. Symbolic data is also far less available than audio data, many symbolic music datasets are created by compiling hand-transcribed music. High-quality automatic transcription remains an unsolved issue.[38][39] However, symbolic music gives more direct access to many higher-level musical features such as chord progressions, melodies, and instrumentation. When working with audio these features first have to be extracted, which often requires working with additional deep learning models. Another consideration to take into account is size. Raw audio is significantly larger than a corresponding digital score. With techniques such as the variational autoencoder [40], audio can be compressed quite considerably for use in machine-learning environments, however, this introduces another costly preprocessing step. In addition, while audio is easier to listen to directly (symbolic data needs to be rendered first), it is difficult to edit once generated.

2.4.2 Tokenisation

Sequences are typically transformed into tokens, a numerical representation of data, to be handled by a machine learning algorithm. Audio-based music generation uses tokenization also as a way to condense audio while retaining its semantic content. Jukebox [15] uses a variational autoencoder[40] with a discretizing bottleneck (VQ-VAE) to create tokens from audio. Musicgen [3] tokenizes audio using the previously developed Encodec model for audio compression, which similarly to VQ-VAE learns a highly condensed discrete representation of audio [41]. These condensed encodings are crucial for generative modeling, also for diffusion models.

2.4.3 Symbolic Tokenisation

Unlike audio, symbolic music is typically already highly condensed, and a variety of different tokenizers exist for different tasks. [42]. Common ways of tokenizing symbolic music align closely with the msical instrument digital interface (MIDI) standard, with individual tokens encoding different midi-events such as note-on, note-off, velocity. The REMI+ [43] tokenisation expands on MIDI-based tokenisation with tokens for bar and position, which is designed

to help capture recurring musical patterns. The PerTok tokenizer designed by Lemonaid[2] encodes micro timings and offsets designed to capture the full spectrum of rhythm in musical performances. These extensions come at a cost, the resulting sequences can become very long, which adversly affects the model[38]. Attempts have been made to condense multiple tokens, such as compound word, or nested tokens.[44]. In MMT [45], tuples of midi-data are combined into single tokens. Our tokenisation varies by task, but in general encodes events into single characters. This is then combined with a byte pair tokenizer, which combines the most common byte-pairs (extracted in a prior run over the input data) into compound tokens until the vocabulary size is exhausted. The vocabulary size indicates the size of the embedding layer, it is orders of magnitude larger than the characters used for the single events.[46]. This approach mimics the compound word approach by condensing the most common combinations into single tokens. This drastically shortens the sequence length by about a third and improves modeling. More in the appendix 4.2

See table **??** in the appendix for a more elaborate display of representations used in symbolic music generators

2.5 Control

Control is an essential aspect of any generative model. Without control, even the best generative models producing beautiful music would be of very limited real-world use. Control allows generative AI tools to become proper collaborative systems, and generate for a wide array of scenarios. In music generation control covers essentially all musical parameters. Parameters vary by representation, symbolic music for instance leaves very little room for any type of timbre control (aside from instrument selection). "Raw" audio models such as Stable Audio (Evans et al., 2024) can for instance be controlled for acoustic settings (i.e jazz music playing in a busy restaurant, in a *large cathedral*, or *through an intercom*), something that is simply not represented in symbolic music. For musical parameters represented in symbolic music, there are different approaches to classifying them. We can differentiate between global and local features [47] or deep vs surface-level features [48].

Global features in music generation encompass high-level descriptors such as genre, function, and instrumentation/orchestration. Global features also incorporate summarizing features, that are calculated from the music itself. As an example, McKay's [49] general-purpose symbolic music classification system defines over 104 global features, 37 of them used for melody descriptors such as probability distributions of note durations, intervals, and pitch classes. These global features can then be used to infer other high-level descriptors such as

[2]https://www.lemonaide.ai/

genre if it's unknown, they can also support music retrieval systems [47]. **Local features** describe individual sequences such as melodic, rhythmic, or harmonic sections. They are more information-rich, which may explain their superior performance in music retrieval [47].

Also relevant in the music generation context are abstract vs concrete features. Music generation can take abstract concepts such as genre [50], [51] or emotion [52] [51] into account, which in turn effects more complex dynamics of concrete features. To illustrate: In Music FaderNets [52] the authors use a variational autoencoder to disentangle the abstract concept of arousal, into several concrete features including rhythmic density, note-density, tempo and dynamic, key. This allows them to change the abstract variable arousal, which cascades into a change of underlying features. This type of disentanglement can aid in situations where labled musical data with the abstract feature is less available, it can also aid in creating more accessible and interpretable generative models.

In the context of this thesis we differentiate between global features, and fine-grained or time-varying features [53]. What is time-varying or global is highly context dependent, a piece may have one time-signature and tempo as is assumed in [51] or it may vary over time as is suggested in [53] or [43]. Other time varying controls could be chords [53][54][55][56], melody [3][56] or texture [56]. For the target application in an MACT - game, time-varying controls are necessary to provide a change in music that triggers a change in the patients improvisation. For a more complete list of the types of controls in many of the common symbolic music generators refer to table **??** in the appendix.

2.5.1 Rhythmic control

The types of control exercised over rhythmic components varies by representation as discussed in 2.4. In CocoMulla [57] generated audio is conditioned with drum tracks and a piano-roll. Similarly in JASCO[58] drums are used for conditioning. In MusicConGen.[55]control for rhythm is added through tracking beats and downbeat. MusicControlNet[54] adds beat and downbeat conditioning to an audio diffusion model. For symbolic systems control of tempo and meter is relatively common [53], [43], [51]. More fine grained control over rhythm is sometimes deployed through note-density (both vertical and horizontal)[53],[59]. Another option is control over texture, which merges harmonic and rhythmic elements. In Polyffusion[56], texture is encoded by a pretrained variational auto-encoder[60]. Another approach [61] involves passing the piano-roll as factor to guide the diffusion process.

2.5.2 Target Features

In order to extend LastMinuteGig with generated music containing appropriate cues we are targeting control of rhythmic patterns. The most simple way of targeting rhythmic patterns is

through note density, implemented as a bar-level instruction of how many notes the following bar should contain. A second simple target feature is note variability (unique notes per bar over number of notes per bar). Both note-density and note-variability are simple time-varying features that are easily extracted, tokenized and introduced to the algorithm.

A potential approach, novel to music generation, would be control mechanisms using either spectral weight or inner metric weight. Here the weight profile is passed as a control mechanism either globally or on a per bar or per section level. *Comment: Introducing the control at a per bar level would be a bit odd since the idea of metric weight is about extracting weights not reflected in the symbolic grid, but it is a sensible way to introduce it to the model in bite-sized chunks able to reflect local change*. Inner metric and spectral weight weight has been successfully applied to dance music classification [62], meter detection [63] and music retrieval [64]. This suggests that it is a powerful feature for explaining various rhythmic aspects, which may make it a good guiding mechanism.

2.6 Non neural net systems

2.6.1 Why look at non-neural systems

While neural net systems currently receive a large amount of attention, other systems are worthy of mention. A recent study [65] performs a comprehensive listening survey comparing different neural net and non-neural net systems. The top-performing systems a Markov Model - MAIA Markov [66] and a deep learning system - MusicTransformer [67] perform similarly well in the listening study. The choice of one of the earliest transformer based music from 2018 [67] and the restiction to symbolic music, given that the study was published in 2023 raises questions as to whether their conclusion that there is no difference in performance between the two approaches still holds, given the rapid advances in the space. However, their criticism is that many deep learning (DL) music generator projects don't look beyond DL and compare their systems based on technical metrics with no obvious impact on how human listeners perceive the systems remain solid. The comparison between neural net and non-neural net systems is worthwhile since deep learning comes with significant downsides relating to explainability and transparency, computational efficiency, as well as copyright and licensing issues regarding their enormous data needs. Additionally, there are hybrid approaches that combine rule-based and deep-learning methods. Those methods being present in research may help researchers and developers maintain a more comprehensive toolkit of techniques and paradigms.

2.6.2 The Illiac suite, introduction to non-neural music generation

Non-neural net systems can be roughly classified as either rule-based or statistical. Both types have been part of some of the earliest attempts at music generation. The 1957 Illiac suites [68][8] hint at many paradigms in music generation for the decades to come. Hiller & Isaacson's first and second movements are generated through encoding rules based on the rules of the first species counterpoint originally published in Fux's 1725 Gradus ad Parnassum[69], which systematically encodes Palestrina's contrapuntal technique. Some rules aim to contain the melody, i.e. limiting the range to an octave, enforcing the same start and end note, and avoiding consecutive melodic jumps. Other rules aim to constrain harmony, such as forbidding parallel octave, fifth, and fourth motion and enforcing consonant harmonies.

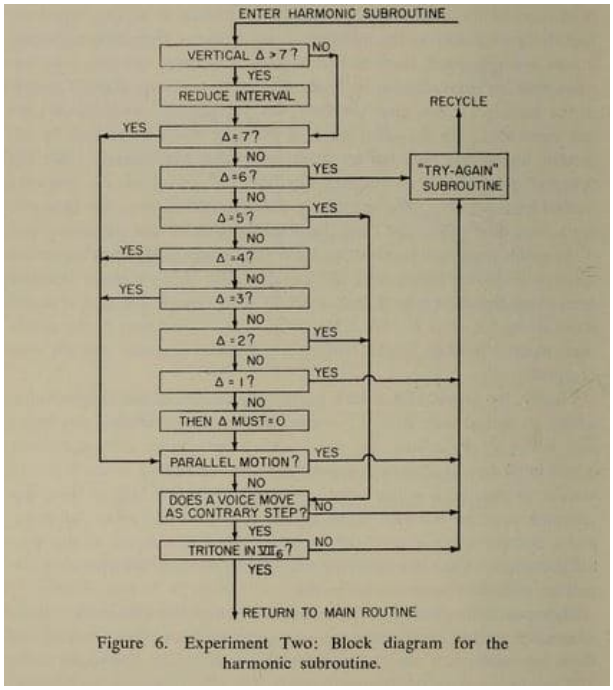

Figure 6. Experiment Two: Block diagram for the harmonic subroutine.

Figure 2.1: Rule-Based - Block Diagram from Hiller and Issacson's book - explaining movement two of the 1957 Illiac Suite.

The third movement automates the creation of 12-tone rows, automating Arnold Schoenberg's 12-tone technique in a semi-randomized fashion. The fourth movement involves using Markov chains with a table of possible intervals stretching from unison to octave, and probabilities assigned to each. Additional restrictions in the latter sections of the movement add memory to the method through higher-order Markov chains that reference previously generated music.

2.6.3 Rule based generation

Centuries of style-defining musicological writing from ancient Mesopotamian tuning charts [70] to Fux's Gradus ad Parnassum [69] and Arnold Schoenberg's 12-tone music have crystallized sets of rules that approximate various styles of music. Many approaches to music generation take advantage of this knowledge and codify it into a computer program creating expert systems for music generation. This can take various forms from simple conditional state machines such as the rules in Hiller & Isaacson's Illiac suite with a handful of conditions to highly complex expert systems such as CHORAL [71] (for which the developer also built a custom programming language) which encodes over 300 rules to realize bach-style chorales from a given melody.

2.6.4 Markov Model Music Generation

Markov models remain a popular method of generating music to this day. At its simplest, a Markov music generator could work off of a transition matrix for pitch classes, such as Richard Pinkerton's 1956 "Banal Melody Generator"[72]. For more complex interactions, Markov chains can be nested or constrained with extensions for global structure [66]. Transition matrices can be built from very little data such as short improvisations used to condition the Continuator [73], but training over a whole corpus is also viable. Other systems configure Markov chains to take additional inputs into account such as Allan, [74] who generates harmonies to given melodies in the style of Bach.

	O	C	D	E	F	G	A	B
O	0.38	0.17	0.10	0.10	0.06	0.13	0.03	0.02
C	0.36	0.23	0.13	0.07	0.02	0.10	0.03	0.07
D	0.26	0.20	0.21	0.19	0.03	0.06	0.01	0.05
E	0.22	0.15	0.18	0.16	0.16	0.12	0.01	0.00
F	0.15	0.00	0.14	0.35	0.14	0.20	0.01	0.01
G	0.29	0.14	0.00	0.16	0.06	0.26	0.08	0.00
A	0.17	0.05	0.07	0.00	0.02	0.36	0.15	0.17
B	0.18	0.30	0.12	0.01	0.01	0.08	0.21	0.08

Figure 2.2: Transition matrix from Pinkerton's 1956 "Banal Music Generator". Probability of a pitch (row) following on a pitch (column), likely pairs are marked in yellow. The probabilities are calculated from a set of 39 nursery rhymes.

2.6.5 Other non-neural music generation

Music generation has also been attempted with other means, such as metaheuristic search for harmony or melody. [75], evolutionary and genetic algorithms[76].

2.6.6 Early Neural Net based systems (up until 2018)

Music generators based on neural nets were introduced as early as 1989. Todd et al. [77] use both a fixed window for a conventional feed-forward neural network but also introduces a feedback loop feeding the network's previous state to the next iteration, foreshadowing future work using RNNs such as [78]. There are also hybrid systems such as HARMONET [79] which merge neural networks with symbolic rule-checking algorithms. More recent RNN based music generators such as FolkRNN [14] use long short term memory (LSTM')s or gated recurrent units (GRU's). Generative Advesarial Networks (GANs) and Variational Autoencoders (VAEs) were and still are popular architectures for music generation. [80]

2.7 State of the art - deep learning for music generation

State-of-the-art music generation, including many commercial applications, leverages the advances in language and image generation achieved in the last five years. Two distinct approaches, namely autoregressive and diffusion-based approaches dominate.

2.7.1 Transformers - sequence modelling without recurrence

Autoregressive music generation is inspired by successful natural language modeling tasks powered by the transformer model. Transformers were originally developed for the task of language translation.[81] They keep the self-attention mechanism which was deployed prior in LSTMs for seq2seq tasks [82] but replace the recurrent connection with positional embeddings and masked attention. This allows the model to train on all tokens in parallel as opposed to one token at a time. At each layer a token is contextialized with the other tokens, via a paralell multi-head attention mechanism the signal of relevant tokens is enhanced, while the signal of other tokens is diminished. The transformer comes in several different configurations. The original transformer contains both encoder and decoder layers - see figure 2.3. Often tasks relating to sequence understanding such as music classification use a encoder only architecture. The BERT series of language models [83] and music understanding models such as MusicBERT[84] are examples. On the other hand sequence generation tasks often employ a decoder only architecturs, this includes the GPT-series [85] and many music generators such as MusicGen[3]. The transformer architecture is the baseline for all large language models currently in use but has also been applied to modalities beyond text including images (they often form the backbone

diffusion models).

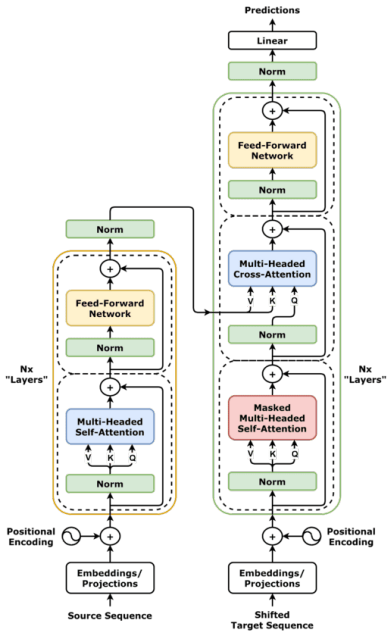

Figure 2.3: Schema of the full transformer encoder-decoder architecture
[a=] By dvgodoy - CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=151216016

2.7.2 Diffusion models - spectrograms and piano-rolls

Diffusion models are widely used for image and audio generation. Diffusion models learn to remove noise from a distribution (i.e an image). Random noise is added to an image, and the model learns to undo this addition. During inference the model starts with random noise (often accompanied by a guiding text prompt) and undoes it until it arrives at a clear image. AudioLDM [86], and StableAudio [87] are diffusion models that operate based on continuous audio encoding by a variational autoencoder. Diffusion models have also been used to generate symbolic music. Polyffusion [56] uses image representations of piano rolls and adapt their diffusion model for various tasks, inpainting, accompaniment and melody generation and generation based on a given chord sequence or texture.

In this thesis we will be targeting transformer models. Transformer models are tremendously popular and behind many of the state of the art results in music generation. There is a tremendous amount of interest in fine-tuning the models, and many methods developed to do so efficiently, including methods to add control. There is a fairly large selection of open-source models that generate music and very good infrastructure to support training and evaluation of those models.

2.8 Implementation of control

Adding control to generative models can be split into three approaches. 1: Choice of architecture and training data, 2: fine-tuning approaches, and 3: external conditioning or guidance.

2.8.1 Control through architecture

The choice of architecture lends itself to different types of control. Transformers are next token predictors, that predict based on the prior sequence. The default training paradigm allows for conditioning with a user defined musical (or audio) sequence. This is true for both audio based models such as MusicGen [3], Jukebox [15] and MusicLM [4] as well as symbolic music models such as MMT [45], MusicTransformer [67] and MusicBERT [84].

Diffusion models are quite flexible compared to transformers, the same model can be used for inpainting, continuation and depending on the representation melody and accompaniment generation through masking.[56][88]

2.8.2 Control through training conditioning

Control can be added through training data. MusicGen[3], a recent text-to-music (audio) transformer is trained on 20000 hours of licensed music from shutterstock and pond5 [3] which includes textual descriptions and tags for genre, tempo, and other factors such as instrumentation. Control is achieved through the joint training of a text description and music. An example description is provided below:

Inspirational dramatic background music! Perfect for trailer, background, advertising, historical film, movie about superheroes, teaser and many other projects![4]

Text-based control, while user-friendly and accessible to non-musicians, is inherently vague. Levels of detail and choice of words vary widely by dataset, even with standardized tags such as genre and tempo. Specialized datasets such as MusicCaps [4], which contains 5500 text music pairs, with 10 seconds of music alongside a free text description and a list of aspect tags created by human experts still suffer substantially from subjectivity [89].

For this reason, the creators of MusicGen [3] add melody conditioning alongside text conditioning and train their model jointly with the chromagram of the melody alongside the text.

In MusicGenStyle [50] perform classifier-free guidance to add style conditioning to MusicGen. They train a music-style encoder that transforms a random subsample of a given reference

[3]https://www.shutterstock.com/music and https://www.pond5.com/

[4]https://www.pond5.com/royalty-free-music/item/95908062-inspiring-dramatic-epic-background-cinematic-music

audio track into tokens that are combined with the embeddings of the text-description. Both the style tokens and text tokens are provided as prefix to the model. The conditioner and the MusicGen transformer are trained jointly on the entire dataset.

The creators of FIGARO[53] enable fine grained control over instrumentation, note density, average pitch and volume on a bar-by-bar basis, in a symbolic music generator through joint conditioning while training.

2.8.3 Adding control through fine-tuning

Both the melody conditioning of MusicGen [3] and the style conditioning of MusicGenStyle [50] retrain the entire MusicGen model on the entire dataset which comes at considerable cost. Fine-tuning or transfer learning is another method through which models can be trained but at considerably smaller cost and using less data. This is particularly useful and widely used in the language domain to adjust large language models for niche use-cases, where the available data may simply not be sufficient to train a large model from scratch. In the examples of MusicGen and MusicGenStyle the availability of data was not a limiting factor since the controlling elements, melody and style can be inferred from the training data. However fine-tuning is also helpful for adding new control mechanisms.

MusiConGen [55] is a fine-tuned variation of MusicGen which adds rhythm and chord control. They propose the jump-finetuning mechanism, where the original model with 1.5 Billion parameters and 48 self-attention layers, is split model into blocks consisting of 4 self-attention layers. They refine the first layer of each block, freezing the remaining layers. Additionally, they apply adaptive in-attention to the first 9 blocks, where the output of the transformer is augmented with copies of the original condition. As a result, only a quarter of the original parameters are tunable, which enables training on consumer GPUs on just 250 hours of music sourced from YouTube (as opposed to 20000 hours). In Coco-Mula [57] the authors adjust a LLAMA adapter with just 4% of parameters, keeping all original MusicGen parameters frozen, and training only the adapter on a small dataset of 300 songs to add drum and chord conditioning.

MuseBarControl [90] is a fine-tuned version of MuseCoco[51] which extends the global controls with fine-grained bar level control for music-generation. They compare several approaches. In the first they augment the prompt (which is generated from text) with additional tokens for bar-wise control of chords, and adjust the loss function to incoperate that. In the second approach they introduce two novel methods, first, they pre-adapt the new parameters (introduced by the lora adapter) to a separate classification task, an auxiliary task . The model classifies whether the a section of music corresponds with the control tokens, the body of the model is trained together with a classification head (which is removed after auxiliary task train-

ing. In the third step they introduce counterfactual loss where the difference in negative log likelyhood conditioned on the original and changed attribute is maximized, which reinforces the models attention to the control. They find that the combination of the three strategies, pre-adaptation on a separate task followed with counterfactual-loss and prompt augmentation yields the strongest model.

2.8.4 Adding control through guidance

There are also other methods that do not involve any finetuning or retraining of the original model. Adding control through rule labels, or control tokens as in [53][55] does require some amount retraining, which is not always feasible, and adding many different types of control may deterioate the model. In these cases, guidance can be used to steer the model towards a certain output. In SMITIN [91] the authors use inference time intervention to guide a model towards a certain output with respect to certain goals. The goals explored are the presence of certain instruments (piano/drums/bass/guitar) in the mix and the quality/realism of the music. The authors train linear probes that learn to associate a state of the attention heads with the stated goal. Then the attention heads are steered in the direction of the probe's output, which increases the probability of the desired quality being present in the generated music.

In Diffusion models, the output is sampled over several steps, at each of these steps it is possible to intervene with guidance to direct the sampling towards a certain goal. In [59], each sampling step is repeated several times, and each time the sample that follows a set of rules most closely is chosen. ControlNet [92] adds spacial control to image generators allowing the guidance of image generation using sketches, poses, edges and depth maps without retraining. MusicControlNet [21] adapts this approach to music generation adding control for time varying factors, melody, dynamics and rhythm.

2.9 Evaluation

How to evaluate generated music is still an open research question. There are no standardized methods according to which evaluations happen[65]. In the context of music generation there are several proposed frameworks to evaluate music. Typically we differentiate between subjective and objective evaluations.

For subjective approaches the methods vary widely [93]. There are simple Turing-type evaluations that test how distinguishable generated and human written music are. Then there are subjective query metrics, where typically likert ratings of different parameters are collected.[56] There are tournament style surveys, where the number of winning pieces are tallied for each approach.[67][53] Finally there are expert evaluations (which can also include likert ratings)

but also analysis of the produced score and musical structure. [14] These evaluations are often paired with statistical hypothesis testing. [53]

Automatic evaluation of generated music include model specific metrics and different musical metrics [93]. Model specific metrics are generic evaluations of a models success to approximate training data, these will vary depending on the model and are not indicative of stylistic success. Examples of this are Negative Log Likelyhood [67], Root Mean Square Error [53] or Perplexity[53]. Musical metrics typically involve comparing a set of generated music to a set of real music, there are plenty of musical similarity measure techniques[94] for a large variety of different use-cases i.e music retrieval, cover, genre and artist detection. A popular comparative metric is calculating the Kulback Leibler (KL) divergence between two datasets with respect to certain metrics i.e count of intervals or unique pitch-classes. However to obtain the divergence one has to select specific features that may only capture a subset of the desired properties. Similar issues arise with other distance metrics i.e cosine similarity, earth movers distance or maximum overlapping area.

Especially in the audio domain, additional AI models are often used for evaluation. MusicGen [3] uses additional classifiers to generate labels for the music and calculates the KL-divergence between the generated labels. Additionally they calculate the Frachet Audio distance, a measure devised to calculate the plausibility of audio (for music enhancement purposes) compared to a large set of studio recordings[95]. Finally they use the CLAP-score which compares the corresponding text description to the latent representation of the generated audio, with text-description of the generated audio with the reference audio. [96]

For this thesis we are interested in two factors, first the plausibility of the generated music, and second the success of the control. How the success of control is evaluated depends on what is controlled for, Examples of controlled parameters and how they are evaluated are as follows:

Note Density. (how many notes per bar). Root mean square error (RME) between generated vs target notes per bar. This is the approach to compare note density used in [53]

Note Variability (number of unique pitch classes, normalized by number onsets) - RME between generated and target.

Rhythmic patterns. Partial Similarity [64] between target and generated music:

Evaluating the plausibility of generated music is more complex, and there is no one method that has been proven superior. Possibly the plausibility of the music will be evaluated with a (small) subjective study. In this scenariou we would follow [45], [97] and [39] and collect likert ratings on questions targeting Coherence Richness Arrangement and Consistency . Theese are would be paired with questions on musical background and preferences of the participant. For

objective rating of the plausibility of the music we will follow the approach by [56] where the KL-divergence between the corpus of generated and a corpus of original music is calculated, most likely KL divergence over a set of extracted features, such as a pitch classes and chords. For a more complete list of different evaluation methods used on symbolic music generators, refer to the notes in the abstract 4.2.

3. Research Plan

3.1 Research Gaps and Opportunities

There are several different methods to add control to music generation models, through training data, architecture choice and through various methods of fine-tuning as discussed in chapter 2 However there are few comparisons in literature between the different methods of adding control, aside from pointing out the general advantages of not retraining the entire model [91][55][57]. Adding control to trained symbolic models is less common in literature as opposed to audio generators. Symbolic music generators tend to be smaller, and are often developed in non-commercial settings, training one from scratch is a often smaller effort. However adding control to a symbolic model presents a unique oppertunity to introduce complex symbolic control mechanisms, while maintaining the flexibility of symbolic music, which can be integrated into most music production environments.
Specifically this thesis investigates adding a certain type of control of rhythm. While generative models feature control of drums [55], tempo, note-density or meter [53], [43], control using inner metric weight, or any complex symbolic characterization of rhythm is novel. However given the success of this characterisation in classification and music-identification tasks it is a promising control feature. The closest approximation is that of texture [56] which has been deployed to some success in diffusion model using VAE disentanglement of harmony and melody. Finally the topic of MACT games offers plenty of opportunity for exploration. Extending the LastMinuteGig[2] application to include generated music, and evaluating the effects of that on enjoyment, engagement and effect is a third area of opportunity. To summarize:

1. Research Opportunity 1: Comparison between different methods of adding control.
2. Research Opportunity 2: Adding Control for rhythmic patterns to a trained symbolic music model
3. Research Opportunity 3: Integration of generated music in MACT application.

Below I outline both the target models and each individual step. The timeline is outlined in the last section, and is quite ambitious. All of the three parts can be decoupled, i.e if the first part is not successful, the second and third parts can still be completed.

3.2 MusicLang Model Suite

For these experiments I am collaborating with the developers of musiclang[1] a suite of open source symbolic music generators who provide the base models. I choose MusicLang for a variety of reasons.

3.2.1 Music Lang Core

Musiclang's core model is a transformer based on LLAMA 2 trained on the Lakh Midi Dataset[98]. Its part of active open source community, well documented and integrated into up to date libraries. If the thesis is successful, I would love to contribute the results here, where they are visible and be of use to others. MusicLang generates high-quality symbolic music, while being relatively small in terms of trainable parameters, fit to run on a consumer machine. The model is already integrated into a custom composition environment, and the developers are activley working to integrate it into a VST plugin to be hosted in the vast majority of music production environments. Finally it already contains controls for chord progression, instrumentation, range, and harmonic coherence, and can generate interpolations, additional instrumental lines and continuations of pre-composed music. This is a useful fallback in case the control of rhythm is not sufficient. There are other control dimensions that can introduce stimulus to the MACT game. For a more complete evaluation of alternative trained symbolic music models see the appendix 4.2.

3.2.2 BassCraft a small transformer model

For the initial experiments comparing different methods of adding control (Opportunity 1) we are using a very small transformer model: BassCraft a GPT2 [85] based transformer. It has an embedding size of 256, 4 attention heads, 4 hidden transformer layers, and a total of 7 million trainable parameters. (For contrast, Llama 3, with the most recent version released in December 2024, comes at different sizes ranging from 1 to 405 billion trainable parameters. While a model like this makes it easier to create a proof of concept due to smaller training costs, there also additional real-world advantages. The model is small and flexible enough to be integrated into a composition environment.

Basscraft is trained to generate a bassline to an existing excerpt of music. It is trained using the lakh-midi dataset [98]. For training, songs with bass-lines are selected (based on the presence of particular MIDI-instrument channels). The tracks are partitioned into snippets between 1 and 16 bars long. The bass-lines are separated from the remaining track and matched as potential output.

[1] https://musiclang.github.io/

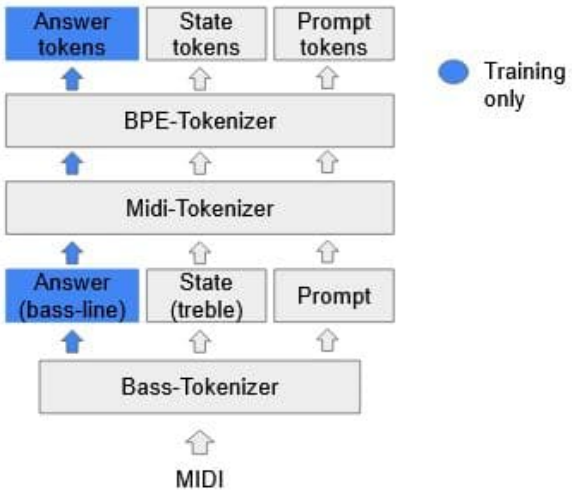

Figure 3.1: Preprocessing/tokenization of the original basscraft model

3.3 Comparing methods of adding control ca 1.5 months

3.3.1 Approach 1 - Vocabulary Extension

Vocabulary expansion is the process to add new vocabulary to a transformer model. The method depends on several factors including which tokenizer is used, the set of new words and how it differs from the prior set. I.e is it a new language with no overlap, or a domain specific subset of a language where you may want to add common domain specific words.

In vocabulary expansion one maintains the prior tokenizer, to keep pretrained knowledge. The tokenizer used for the Basscraft model is a modified byte pair encoding (BPE) tokenizer [46], which is trained starting from a general alphabet, and greedily merges the most common pairs of tokens until the target vocabulary size is reached (16000). It is difficult to simply add new tokens here, as it would require retraining the BPE tokenizer, which will transform the embedding layer making it unusable. Common approaches are merging tokenizers, resizing the vocabulary to add new tokens while keeping the prior embeddings intact or replacing vocabulary items. We choose vocabulary replacement of unused tokens with the new control tokens. This will inevitibally extend the sequence size, since the new tokens are not merged in the BPE process, which may adversly effect the model performance. However this will allow us to maintain the same embedding size, while still benefiting from the model's prior training.

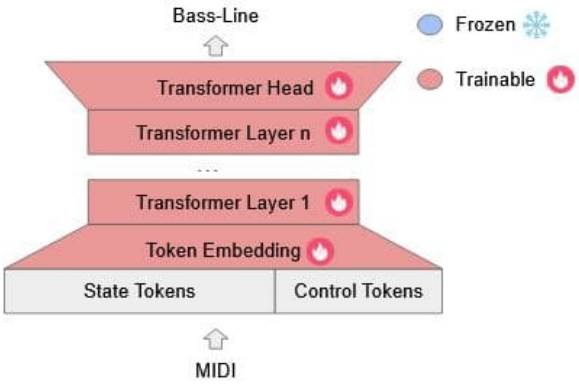

Figure 3.2: Vocabulary transfer with full fine tuning

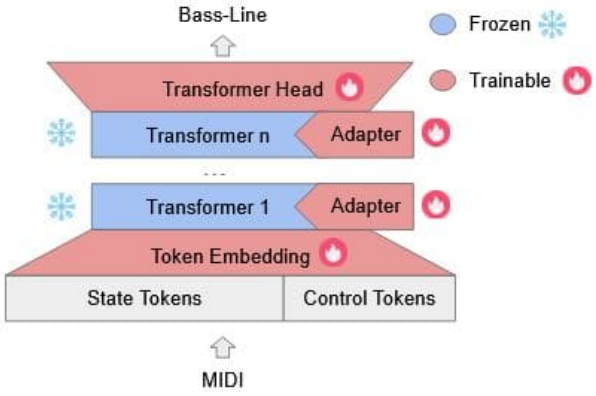

Figure 3.3: Vocabulary transfer with parameter efficient fine tuning

3.3.2 Approach 2 - Integrating of control tokens

In this experiment we adjust the approach used in Coco-Mulla [57] inserting the embedded condition prefix into the last frozen layers of the model. The embedding is learnable. In contrast to the previous method we are not working with tokens as input tokens, we are incorporating them into the model after c layers. This method has the benifit that we don't need to worry about tokenisation and the vocabulary of the original model, as those remain unchanged. The difficulty is learning the embedding of the control tokens.

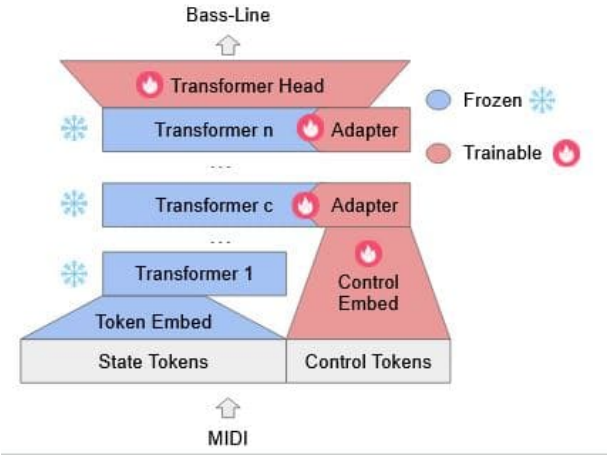

Figure 3.4: Integration of control tokens with parameter efficient fine tuning

3.3.3 Approach 3 - Post Hoc Conditioning

This is the approach used by SMITIN[91]. ->

One potential problem: This may be very difficult to implement. SMITIN only has binary scenarios (presence of certain instruments), our current target variables are larger. What is nice about this approach is that it is very flexible, can be used in conjunction with other methods, and roughly emulates the guidance mechanisms used in diffusion models.

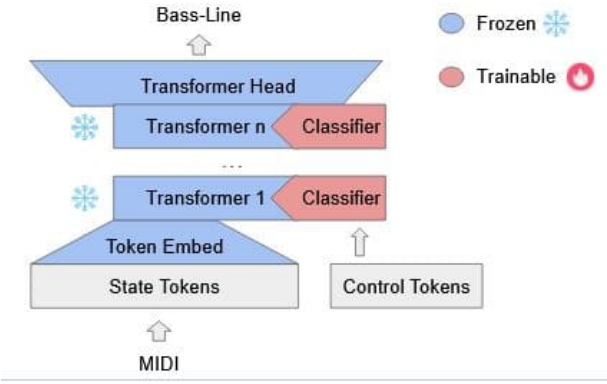

Figure 3.5: Integration of control using inference time interference

If these experiments are not successful, we can follow the approach of [90] and try additional training using auxiliary tasks and using a modified (counterfactual) loss function.

Other potential avenues to explore in this section (if time remains) is how controls stack. Adapters for the different target controls can be trained separately, then merged.

3.4 Application to larger model 1.5 months

The first step in this experiment is extracting and tokenizing either spectral or inner metric weight which is then tokenized and passed to the MusicLang Llama 2 model. They will be integrated with the most successful method determined above.

3.5 Integration into LastMinuteGig 2 months

This project is motivated by extending the music engine of LastMinuteGig. [2]. The application currently employs a very simple algorithm. I will outline it below.

- 1: Randomly choose key, rhythmic pattern, chord progression and tempo from a pool of possible values.
- 2: Play corresponding percussion audio clip (depends on tempo and rhythmic pattern)
- 3: When user plays the button trigger the guitar sound of the current chord.
- 4: After 8 bars - make a random change (to rhythm, tempo, pause)

This music engine will be changed in the following way. Most of the generative process will be done asynchronously.

- 1: Create $n = 100$ musical pieces with different control settings
- 2: For each piece
- 3: Extract chords from prompt (assuming high control success) - save mapping $m1$
- 4: Extract change of rhythmic pattern (or tempo or meter or instrumentation since these are already controllable in the musiclang model) - save mapping
- 5: Render symbolic music to audio

During game play

- 1: Load random audio clip and mappings.
- 2: Play associated guitar chords when player taps.
- 3: Register changes in tapping on musical changes.
- 4: Repeat when audio clip has finished playing.

This will be followed by a user study on healthy participants following the original protocol[2]

3.6 Thesis Timeline

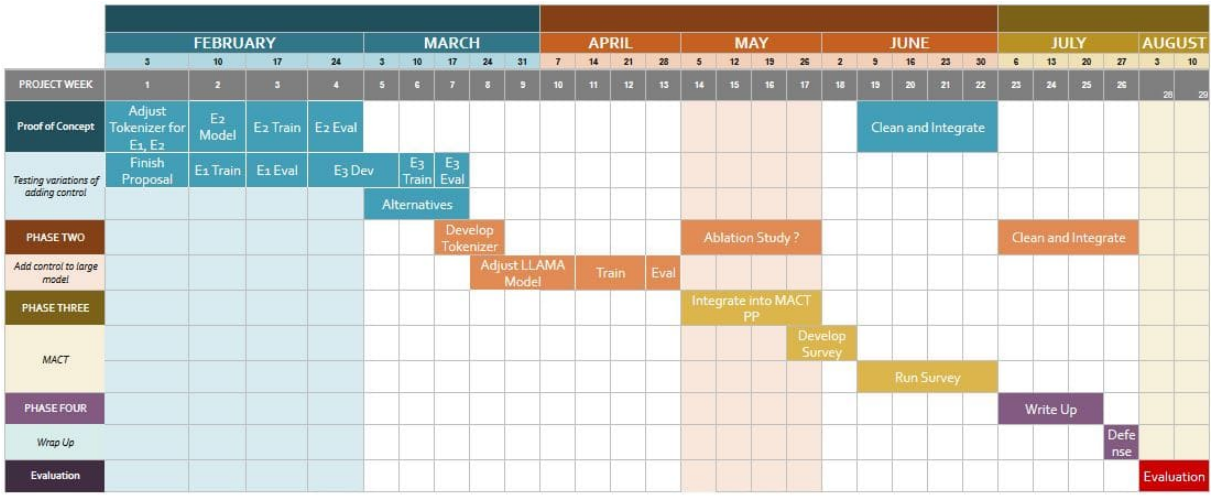

Figure 3.6: Thesis project plan

Bibliography

[1] K. Rogers and M. Weber, "Audio habits and motivations in video game players," in *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound*, ser. AM '19, New York, NY, USA: Association for Computing Machinery, Sep. 2019, pp. 45–52, ISBN: 978-1-4503-7297-8. DOI: 10.1145/3356590.3356599. [Online]. Available: https://doi.org/10.1145/3356590.3356599.

[2] E. Chalkiadakis, "Developing and evaluating a musical attention control training computer game application.," M.S. thesis, Utrecht University, Utrecht, NL, Jan. 2022. [Online]. Available: https://studenttheses.uu.nl/bitstream/handle/20.500.12932/500/Master%20Thesis%20-%20Developing%20and%20evaluating%20a%20Musical%20Attention%20Control%20Training%20computer%20game%20application%20%285892252%29.pdf?sequence=1%5C&isAllowed=y.

[3] J. Copet, F. Kreuk, I. Gat, *et al.*, "Simple and controllable music generation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. DOI: 10.48550/arXiv.2306.05284. [Online]. Available: https://openreview.net/forum?id=jtiQ26sCJi.

[4] A. Agostinelli, T. I. Denk, Z. Borsos, *et al.*, "Musiclm: Generating music from text," *arXiv*, no. arXiv:2301.11325, Jan. 2023. DOI: 10.48550/arXiv.2301.11325. [Online]. Available: http://arxiv.org/abs/2301.11325.

[5] B. R. Ferdinand Meyen, ""verknallt in einen talahon": So reagiert der talahon-produzent auf die rassismus-vorwürfe," *Bayern 2 Zuendfunk*, Aug. 2024. [Online]. Available: https://www.br.de/radio/bayern2/sendungen/zuendfunk/verknallt-in-einen-talahon-produzent-rassismus-vorwuerfe-charts-udio-ki-100.html.

[6] M. Stassen, "Suno, with a $500m valuation, has admitted training its ai on copyrighted music," *Music Business Worldwide*, Apr. 2024. [Online]. Available: https://www.musicbusinessworldwide.com/suno-with-a-500m-valuation-has-admitted-training-its-ai-on-copyrighted-music-it-just-named-timbaland-as-a-strategic-advisor1/.

[7] A. France-Presse, "First recording of computer-generated music – created by alan turing – restored," *The Guardian*, Sep. 2016, ISSN: 0261-3077. [Online]. Available: https://www.theguardian.com/science/2016/sep/26/first-recording-computer-generated-music-created-alan-turing-restored-enigma-code.

[8] L. Hiller and L. M. (M. Isaacson, *Experimental music; composition with an electronic computer*. New York, McGraw-Hill, 1959. [Online]. Available: http://archive.org/details/experimentalmusi00hill.

[9] I. Xenakis, *Formalized Music Thought and Mathematics in Composition*. Pendragon Press, 1992. [Online]. Available: https://monoskop.org/images/7/74/Xenakis_Iannis_Formalized_Music_Thought_and_Mathematics_in_Composition.pdf.

[10] D. Cope, *New directions in music*. Dubuque, Iowa : W.C. Brown, 1989, ISBN: 978-0-697-03342-0. [Online]. Available: http://archive.org/details/unset0000unse_x4d7.

[11] E. Featherstone, "Introducing the next generation of music makers," *The Guardian*, Aug. 2017, ISSN: 0261-3077. [Online]. Available: https://www.theguardian.com/small-business-network/2017/aug/29/computer-write-music-jukedeck-artificial-intelligence.

[12] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," in *18th International Society*

for Music Information Retrieval Conference (ISMIR), Suzhou, China, Jul. 2017. DOI: 10.48550/arXiv.1703.10847. [Online]. Available: http://arxiv.org/abs/1703.10847.

[13] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: A steerable model for bach chorales generation," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, PMLR 70:1362-1371, Jun. 2017. DOI: 10.48550/arXiv.1612.01010. [Online]. Available: http://arxiv.org/abs/1612.01010.

[14] B. L. T. Sturm and O. Ben-Tal, "Folk the algorithms: (mis)applying artificial intelligence to folk music," in E. R. Miranda, Ed. Cham: Springer International Publishing, 2016, pp. 423–454, ISBN: 978-3-030-72115-2. DOI: 10.1007/978-3-030-72116-9_16. [Online]. Available: https://link.springer.com/10.1007/978-3-030-72116-9_16.

[15] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv*, no. arXiv:2005.00341, Apr. 2020. DOI: 10.48550/arXiv.2005.00341. [Online]. Available: http://arxiv.org/abs/2005.00341.

[16] P. Christine, *Musenet*, Apr. 2019. [Online]. Available: openai.com/blog/musenet.

[17] en. [Online]. Available: https://suno.com/about.

[18] M. E. Malandro, "Composer's assistant: An interactive transformer for multi-track midi infilling," in *2024 International Society for Music Information Retrieval*, arXiv:2301.12525 [cs], San Francisco, USA: ISMIR, Jul. 2023. DOI: 10.48550/arXiv.2301.12525. [Online]. Available: http://arxiv.org/abs/2301.12525.

[19] C. Plut and P. Pasquier, "Generative music in video games: State of the art, challenges, and prospects," *Entertainment Computing*, vol. 33, p. 100 337, Mar. 2020, ISSN: 1875-9521. DOI: 10.1016/j.entcom.2019.100337.

[20] K. Worrall and T. Collins, "Considerations and concerns of professional game composers regarding artificially intelligent music technology," *IEEE Transactions on Games*, vol. 16, pp. 586–597, Sep. 2024, ISSN: 2475-1502, 2475-1510. DOI: 10.1109/TG.2023.3319085.

[21] K. R. Agres, R. S. Schaefer, A. Volk, *et al.*, "Music, computing, and health: A roadmap for the current and future roles of music technology for health care and well-being," *Music & Science*, vol. 4, p. 2 059 204 321 997 709, Jan. 2021, ISSN: 2059-2043. DOI: 10.1177/2059204321997709.

[22] J.-K. Park and S. J. Kim, "Dual-task-based drum playing with rhythmic cueing on motor and attention control in patients with parkinson's disease: A preliminary randomized study," eng, *International Journal of Environmental Research and Public Health*, vol. 18, no. 19, p. 10 095, Sep. 2021, ISSN: 1660-4601. DOI: 10.3390/ijerph181910095.

[23] M. Martin-Moratinos, M. Bella-Fernández, and H. Blasco-Fontecilla, "Effects of music on attention-deficit/hyperactivity disorder (adhd) and potential application in serious video games: Systematic review," *Journal of Medical Internet Research*, vol. 25, e37742, May 2023, ISSN: 1439-4456. DOI: 10.2196/37742.

[24] V. Pasiali, A. B. LaGasse, and S. L. Penn, "The effect of musical attention control training (mact) on attention skills of adolescents with neurodevelopmental delays: A pilot study," *Journal of Music Therapy*, vol. 51, no. 4, pp. 333–354, 2014, ISSN: 0022-2917. DOI: 10.1093/jmt/thu030.

[25] R. van Alphen, G. J. J. M. Stams, and L. Hakvoort, "Musical attention control training for psychotic psychiatric patients: An experimental pilot study in a forensic psychiatric hospital," *Frontiers in Neuroscience*, vol. 13, p. 570, 2019, ISSN: 1662-4548. DOI: 10.3389/fnins.2019.00570.

[26] D. Tencer, "New ai-powered 'instant' music-making app udio raises $10m; launches with backing from will.i.am, common, unitedmasters, a16z," en-US, *Music Business Worldwide*, Apr. 2024. [Online]. Available: https://www.musicbusinessworldwide.com/new-ai-powered-instant-music-making-app-udio-raises-10m-launches-with-backing-from-will-i-am-common-unitedmasters-a16z/.

[27] M. M. Kaba, M. N. River, and A. R. Perry, *Demand for jury trial*, Jun. 2024. [Online]. Available: https://storage.courtlistener.com/recap/gov.uscourts.mad.272063/gov.uscourts.mad.272063.1.0.pdf.

[28] M. Duan, A. Suri, N. Mireshghallah, *et al.*, "Do membership inference attacks work on large language models?" In *Conference on Language Modeling (COLM)*, arXiv:2402.07841 [cs], arXiv, Sep. 2024. DOI: 10.48550/arXiv.2402.07841. [Online]. Available: http://arxiv.org/abs/2402.07841.

[29] N. Carlini, J. Hayes, M. Nasr, *et al.*, "Extracting training data from diffusion models," in *Proceedings of the 32nd USENIX Security Symposium*, arXiv:2301.13188 [cs], Anaheim, CA, USA: arXiv, Jan. 2023. DOI: 10.48550/arXiv.2301.13188. [Online]. Available: http://arxiv.org/abs/2301.13188.

[30] E. Newton-Rex, "Suno is a music ai company aiming to generate $120 billion per year. but is it trained on copyrighted recordings?" en-US, *Music Business Worldwide*, Apr. 2024. [Online]. Available: https://www.musicbusinessworldwide.com/suno-is-a-music-ai-company-aiming-to-generate-120-billion-per-year-newton-rex/.

[31] R. Reed, "Does chatgpt violate new york times' copyrights?" en-us, *Harvard Law School*, May 2024. [Online]. Available: https://hls.harvard.edu/today/does-chatgpt-violate-new-york-times-copyrights/.

[32] *Copyright law of the united states title 17, chapter 1, section 107*. [Online]. Available: https://www.copyright.gov/title17/92chap1.html#107.

[33] M. Singh, *Indian filmmaker ram gopal varma abandons human musicians for ai-generated music*, en-US, Sep. 2024. [Online]. Available: https://techcrunch.com/2024/09/19/indian-filmmaker-ram-gopal-varma-abandons-human-musicians-for-ai-generated-music/.

[34] M. Stassen, ""there are now 120,000 new tracks hitting music streaming services each day"," *Music Business Worldwide*, May 2023. [Online]. Available: https://www.musicbusinessworldwide.com/there-are-now-120000-new-tracks-hitting-music-streaming-services-each-day/.

[35] A. Marechal, "Generative ai: Energy consumption soars," en-GB, *Polytechnique Insights*, Nov. 2024. [Online]. Available: https://www.polytechnique-insights.com/en/columns/energy/generative-ai-energy-consumption-soars/.

[36] J. Kaplan, S. McCandlish, T. Henighan, *et al.*, "Scaling laws for neural language models," *arXiv*, no. arXiv:2001.08361, Jan. 2020, arXiv:2001.08361 [cs]. DOI: 10.48550/arXiv.2001.08361. [Online]. Available: http://arxiv.org/abs/2001.08361.

[37] A. Holzapfel, A.-K. Kaila, and P. Jääskeläinen, "Green mir? investigating computational cost of recent music-ai research in ismir," in *25th Int. Society for Music Information Retrieval Conf*, San Francisco, United States, 2024. [Online]. Available: https://drive.google.com/file/d/1rAepoJk1U2R4g3S_AcxdqWDGzRGd7xPg/view?usp=embed_facebook.

[38] S. Ji, X. Yang, and J. Luo, "A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges," en, *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–39, Jan. 2024, ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3597493.

[39] H. Chen, J. B. L. Smith, J. Spijkervet, *et al.*, "Sympac: Scalable symbolic music generation with prompts and constraints," in *25th Int. Society for Music Information Retrieval Conference*, arXiv:2409.03055 [cs], San Fancisco, CA, USA, Sep. 2024. DOI: 10.48550/arXiv.2409.03055. [Online]. Available: http://arxiv.org/abs/2409.03055.

[40] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, arXiv:1312.6114 [stat], arXiv, 2014. DOI: 10.48550/arXiv.1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114.

[41] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," en, *Transactions on Machine Learning Research*, 2023, arXiv:2210.13438 [eess]. DOI: 10.48550/arXiv.2210.13438. [Online]. Available: http://arxiv.org/abs/2210.13438.

[42] N. Fradet, J.-P. Briot, and F. Chhel, "Miditok: A python package for midi file tokenization," en, *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021)*, 2021.

[43] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20, New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 1180–1188, ISBN: 978-1-4503-7988-5. DOI: 10.1145/3394171.3413671. [Online]. Available: https://doi.org/10.1145/3394171.3413671.

[44] J. Ryu, H.-W. Dong, J. Jung, and D. Jeong, "Nested music transformer: Sequentially decoding compound tokens in symbolic music and audio generation," in *25th Int. Society for Music Information Retrieval Conference (ISMIR 2024)*, arXiv:2408.01180 [cs], Aug. 2024. DOI: 10.48550/arXiv.2408.01180. [Online]. Available: http://arxiv.org/abs/2408.01180.

[45] H.-W. Dong, K. Chen, S. Dubnov, J. McAuley, and T. Berg-Kirkpatrick, "Multitrack music transformer," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, arXiv:2207.06983, arXiv, May 2023. DOI: 10.48550/arXiv.2207.06983. [Online]. Available: http://arxiv.org/abs/2207.06983.

[46] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Association for Computational Linguistics (ACL 2016)*, arXiv:1508.07909 [cs], Berlin, Germany: arXiv, Jun. 2016. DOI: 10.48550/arXiv.1508.07909. [Online]. Available: http://arxiv.org/abs/1508.07909.

[47] P. Van Kranenburg, A. Volk, and F. Wiering, "A comparison between global and local features for computational classification of folk song melodies," en, *Journal of New Music Research*, vol. 42, no. 1, pp. 1–18, Mar. 2013, ISSN: 0929-8215, 1744-5027. DOI: 10.1080/09298215.2012.718790.

[48] J. Blacking, "Deep and surface structures in venda music," *Yearbook of the International Folk Music Council*, vol. 3, pp. 91–108, 1971, ISSN: 0316-6082. DOI: 10.2307/767458.

[49] C. McKay, "Automatic genre classification using large high-level musical feature sets.," en, Ph.D. dissertation, McGill, Montreal QC Canada, 2004. [Online]. Available: https://www.researchgate.net/publication/220723648_Automatic_Genre_Classification_Using_Large_High-Level_Musical_Feature_Sets.

[50] S. Rouard, Y. Adi, J. Copet, A. Roebel, and A. Défossez, "Audio conditioning for music generation via discrete bottleneck features," in *25th Int. Society for Music Information Retrieval Conference*, arXiv:2407.12563, San Fancisco, CA, USA: arXiv, Jul. 2024. DOI: 10.48550/arXiv.2407.12563. [Online]. Available: http://arxiv.org/abs/2407.12563.

[51] P. Lu, X. Xu, C. Kang, *et al.*, "Musecoco: Generating symbolic music from text," *arXiv*, no. arXiv:2306.00110, May 2023, arXiv:2306.00110 [cs]. DOI: 10.48550/arXiv.2306.00110. [Online]. Available: http://arxiv.org/abs/2306.00110.

[52] H. Tan and D. Herremans, "Music fadernets: Controllable music generation based on high-level features via low-level feature modelling," en, in *Proc. of 21st International Society of Music Information Retrieval Conference, ISMIR 2020*, Montreal QC Canada, 2020. [Online]. Available: https://www.researchgate.net/publication/343500818_Music_FaderNets_Controllable_Music_Generation_Based_On_High-Level_Features_via_Low-Level_Feature_Modelling.

[53] D. v. Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, "Figaro: Generating symbolic music with fine-grained artistic control," *International Conference on Learning Representations*, 2023, arXiv:2201.10936. DOI: 10.48550/arXiv.2201.10936. [Online]. Available: http://arxiv.org/abs/2201.10936.

[54] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," en, *arXiv*, no. arXiv:2311.07069, Nov. 2023, arXiv:2311.07069 [cs]. [Online]. Available: http://arxiv.org/abs/2311.07069.

[55] Y.-H. Lan, W.-Y. Hsiao, H.-C. Cheng, and Y.-H. Yang, "Musicongen: Rhythm and chord control for transformer-based text-to-music generation," en, in *25th Int. Society for Music Information Retrieval Conference*, arXiv:2407.15060 [cs], San Fancisco, CA, USA: arXiv, Jul. 2024. [Online]. Available: http://arxiv.org/abs/2407.15060.

[56] L. Min, J. Jiang, G. Xia, and J. Zhao, "Polyffusion: A diffusion model for polyphonic score generation with internal and external controls," in *24th Int. Society for Music Information Retrieval Conference (ISMIR 2023)*, arXiv:2307.10304 [cs], Milan, Italy: arXiv, Jul. 2023. [Online]. Available: http://arxiv.org/abs/2307.10304.

[57] L. Lin, G. Xia, J. Jiang, and Y. Zhang, "Content-based controls for music large language modeling," en, *arXiv*, no. arXiv:2310.17162, Oct. 2024, arXiv:2310.17162 [cs]. DOI: 10.48550/arXiv.2310.17162. [Online]. Available: http://arxiv.org/abs/2310.17162.

[58] O. Tal, A. Ziv, I. Gat, F. Kreuk, and Y. Adi, "Joint audio and symbolic conditioning for temporally controlled text-to-music generation," en, in *25th Int. Society for Music Information Retrieval Conference*, arXiv:2406.10970 [cs], San Fancisco, CA, USA, Jun. 2024. [Online]. Available: http://arxiv.org/abs/2406.10970.

[59] Y. Huang, A. Ghatare, Y. Liu, *et al.*, "Symbolic music generation with non-differentiable rule guided diffusion," in *The Forty-First International Conference on Machine Learning*, arXiv:2402.14285 [cs], Vienna, Austria: arXiv, Sep. 2024. DOI: 10.48550/arXiv.2402.14285. [Online]. Available: http://arxiv.org/abs/2402.14285.

[60] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *21st International Conference on Music Information Retrieval (ISMIR 2020)*, arXiv:2008.07122 [cs], Montreal QC Canada: arXiv, Aug. 2020. DOI: 10.48550/arXiv.2008.07122. [Online]. Available: http://arxiv.org/abs/2008.07122.

[61] T. Zhu, H. Liu, Z. Jiang, and Z. Zheng, "Symbolic music generation with fine-grained interactive textural guidance," en, *arXiv*, no. arXiv:2410.08435, Oct. 2024, arXiv:2410.08435 [cs]. DOI: 10.48550/arXiv.2410.08435. [Online]. Available: http://arxiv.org/abs/2410.08435.

[62] E. Chew, A. Volk, and C.-Y. Lee, "Dance music classification using inner metric analysis," en, in *The Next Wave in Computing, Optimization, and Decision Technologies*, B. Golden, S. Raghavan, and E. Wasil, Eds., Boston, MA: Springer US, 2005, pp. 355–370, ISBN: 978-0-387-23529-5. DOI: 10.1007/0-387-23529-9_23.

[63] W. B. Haas and A. Volk, "Meter detection in symbolic music using inner metric analysis," in *International Society for Music Information Retrieval Conference (ISMIR 2016)*, Aug. 2016. [Online]. Available: https://www.semanticscholar.org/paper/Meter-Detection-in-Symbolic-Music-Using-Inner-Haas-Volk/8a72fc0874e0476f3dc0b92a109ebf854a4551cc.

[64] A. Volk, J. Garbers, P. Van Kranenburg, F. Wiering, L. Grijp, and R. C. Veltkamp, "Comparing computational approaches to rhythmic and melodic similarity in folksong research," en, in *Mathematics and Computation in Music* (Communications in Computer and Information Science), T. Klouche and T. Noll, Eds., Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 37, pp. 78–87, ISBN: 978-3-642-04578-3. DOI: 10.1007/978-3-642-04579-

0_8. [Online]. Available: http://link.springer.com/10.1007/978-3-642-04579-0_8.

[65] Z. Yin, F. Reuben, S. Stepney, and T. Collins, "Deep learning's shallow gains: A comparative evaluation of algorithms for automatic music generation," en, *Machine Learning*, vol. 112, no. 5, pp. 1785–1822, May 2023, ISSN: 1573-0565. DOI: 10.1007/s10994-023-06309-w.

[66] T. Collins and R. Laney, "Computer-generated stylistic compositions with long-term repetitive and phrasal structure," None, *Journal of Creative Music Systems*, vol. 1, no. 22, Mar. 2017, ISSN: 2399-7656. DOI: 10.5920/JCMS.2017.02. [Online]. Available: https://www.jcms.org.uk/article/id/510/.

[67] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, *et al.*, "Music transformer," *arXiv*, no. arXiv:1809.04281, Dec. 2018, arXiv:1809.04281. DOI: 10.48550/arXiv.1809.04281. [Online]. Available: http://arxiv.org/abs/1809.04281.

[68] T. Funk, "A musical suite composed by an electronic brain: Reexamining the illiac suite and the legacy of lejaren a. hiller jr.," *Leonardo Music Journal*, vol. 28, pp. 19–24, 2018, ISSN: 0961-1215.

[69] J. J. Fux, *Gradus ad Parnassum*. Vienna, Austria: Catholicae Majestati Aulae-Typographi, 1725. [Online]. Available: https://www.loc.gov/item/16002789/.

[70] S. Mirelman, "Tuning procedures in ancient iraq," en, *Analytical Approaches to World Music 2*, no. 2, pp. 43–56, 2013.

[71] K. Ebcioğlu, "An expert system for harmonizing chorales in the style of j. s. bach," *Understanding Music with AI: Perspectives on Music Cognition*, vol. eds. M. Balaban, K. Ebcioğlu, and O. Laske, pp. 145–185, 1994.

[72] R. C. Pinkerton, "Information theory and melody," *Scientific American*, vol. 194, no. 2, pp. 77–87, 1956.

[73] F. Pachet, "The continuator: Musical interaction with style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, Sep. 2003, ISSN: 0929-8215. DOI: 10.1076/jnmr.32.3.333.16861.

[74] M. Allan, "Harmonising chorales in the style of johann sebastian bach," en, Ph.D. dissertation, Edinborough, 2002. [Online]. Available: https://www.researchgate.net/publication/238242337_Harmonising_Chorales_in_the_Style_of_Johann_Sebastian_Bach.

[75] E. V. Altay and B. Alatas, "Music based metaheuristic methods for constrained optimization," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS 2018)*, Mar. 2018, pp. 1–6. DOI: 10.1109/ISDFS.2018.8355355. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8355355.

[76] J. Polito, J. M. Daida, and T. F. Bersano-Begey, "Musica ex machina: Composing 16th-century counterpoint with genetic programming and symbiosis," *Evolutionary Programming*, Lecture Notes in Computer Science, vol. 1213, P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, Eds., pp. 113–123, 1997, Book Title: Evolutionary Programming VI. DOI: 10.1007/BFb0014805.

[77] P. M. Todd, "A connectionist approach to algorithmic composition," *Computer Music Journal*, vol. 13, no. 4, pp. 27–43, 1989, ISSN: 0148-9267. DOI: 10.2307/3679551.

[78] M. C. Mozer, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing," *Connection Science*, vol. 6, no. 2–3, pp. 247–280, Jan. 1994, ISSN: 0954-0091. DOI: 10.1080/09540099408915726.

[79] H. Hild, J. Feulner, and W. Menzel, "Harmonet: A neural net for harmonizing chorales in the style of l.s.bach," en, *Advances in Neural Information Processing Systems 4 (NIPS 1991)*, 1991.

[80] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona, "A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends," *Expert Systems with Applications*, vol. 209, p. 118 190, Dec. 2022, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2022.118190.

[81] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, arXiv, 2017. DOI: 10.48550/arXiv.1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[82] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," en, *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Dec. 2014, arXiv:1409.3215 [cs]. DOI: 10.48550/arXiv.1409.3215. [Online]. Available: http://arxiv.org/abs/1409.3215.

[83] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," en, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (ACL 2019)*, arXiv:1810.04805 [cs], arXiv, May 2019. DOI: 10.48550/arXiv.1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.

[84] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, "Musicbert: Symbolic music understanding with large-scale pre-training," en, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 791–800, Jun. 2021, arXiv:2106.05630 [cs]. DOI: 10.18653/v1/2021.findings-acl.70.

[85] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:160025533.

[86] H. Liu, Z. Chen, Y. Yuan, *et al.*, "Audioldm: Text-to-audio generation with latent diffusion models," in *Proceedings of the 40th International Conference on Machine Learning (PMLR)*, arXiv:2301.12503 [cs], arXiv, Sep. 2023. DOI: 10.48550/arXiv.2301.12503. [Online]. Available: http://arxiv.org/abs/2301.12503.

[87] Z. Evans, J. D. Parker, C. J. Carr, Z. Zukowski, J. Taylor, and J. Pons, "Stable audio open," *arXiv*, no. arXiv:2407.14358, Jul. 2024, arXiv:2407.14358 [cs]. DOI: 10.48550/arXiv.2407.14358. [Online]. Available: http://arxiv.org/abs/2407.14358.

[88] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

[89] M. Lee, S. Doh, and D. Jeong, "Annotator subjectivity in the musiccaps dataset," en, *HCMIR23: 2nd Workshop on Human-Centric Music Information Research)*, vol. 1, Nov. 2023.

[90] Y. Shu, H. Xu, Z. Zhou, A. v. d. Hengel, and L. Liu, "Musebarcontrol: Enhancing fine-grained control in symbolic music generation through pre-training and counterfactual loss," *arXiv*, no. arXiv:2407.04331, Jul. 2024, arXiv:2407.04331 [cs]. DOI: 10.48550/arXiv.2407.04331. [Online]. Available: http://arxiv.org/abs/2407.04331.

[91] J. Koo, G. Wichern, F. G. Germain, S. Khurana, and J. L. Roux, "Smitin: Self-monitored inference-time intervention for generative music transformers," *IEEE Open Journal of Signal Processing 2025*, Apr. 2024, arXiv:2404.02252. DOI: 10.48550/arXiv.2404.02252. [Online]. Available: http://arxiv.org/abs/2404.02252.

[92] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *IEEE/CVF International Conference on Computer Vision (ICCV 2023)*,

arXiv:2302.05543 [cs], arXiv, Nov. 2023. DOI: 10.1109/ICCV51070.2023.00355. [Online]. Available: http://arxiv.org/abs/2302.05543.

[93] Z. Xiong, W. Wang, J. Yu, Y. Lin, and Z. Wang, "A comprehensive survey for evaluation methodologies of ai-generated music," *arXiv*, no. arXiv:2308.13736, Aug. 2023, arXiv:2308.13736 [cs]. DOI: 10.48550/arXiv.2308.13736. [Online]. Available: http://arxiv.org/abs/2308.13736.

[94] K. Gurjar and Y.-S. Moon, "A comparative analysis of music similarity measures in music information retrieval systems," en, *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 32–55, Feb. 2018. DOI: 10.3745/JIPS.04.0054.

[95] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A metric for evaluating music enhancement algorithms," en, *INTERSPEECH 2019*, Jan. 2019, arXiv:1812.08466 [eess]. DOI: 10.48550/arXiv.1812.08466. [Online]. Available: http://arxiv.org/abs/1812.08466.

[96] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "Clap: Learning audio concepts from natural language supervision," en, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, arXiv:2206.04769 [cs], arXiv, 2023. DOI: 10.48550/arXiv.2206.04769. [Online]. Available: http://arxiv.org/abs/2206.04769.

[97] B. Yu, P. Lu, R. Wang, *et al.*, "Museformer: Transformer with fine- and coarse-grained attention for music generation," en, in *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)*, arXiv:2210.10349 [cs], arXiv, Oct. 2022. DOI: 10.48550/arXiv.2210.10349. [Online]. Available: http://arxiv.org/abs/2210.10349.

[98] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," en, Ph.D. dissertation, Columbia University, New York, NY, USA, 2016.

[99] *Canonical jsb dataset chorals 389*. [Online]. Available: https://github.com/czhuang/JSB-Chorales-dataset.

[100] J. Keith, *Thesession data*. [Online]. Available: https://github.com/adactio/TheSession-data.

[101] C. Hawthorne, A. Stasyuk, A. Roberts, *et al.*, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7.

[102] *Hook theory collection of pop-song transcriptions*. [Online]. Available: https://www.hooktheory.com/theorytab.

[103] Z. Wang, K. Chen, J. Jiang, *et al.*, "Pop909: A pop-song dataset for music arrangement generation," en, in *21st International Conference on Music Information Retrieval (ISMIR 2020)*, 2020.

[104] L. Crestel, P. Esling, L. Heng, and S. McAdams, "A database linking piano and orchestral midi scores with application to automatic projective orchestration," in *18th International Society for Music Information Retrieval Conference*, arXiv:1810.08611 [cs], Suzhou, China: arXiv, Oct. 2018. DOI: 10.48550/arXiv.1810.08611. [Online]. Available: http://arxiv.org/abs/1810.08611.

[105] *Classical archives dataset donation*. [Online]. Available: https://www.classicalarchives.com/newca/#!/.

[106] *Bit midi dataset donation*. [Online]. Available: https://bitmidi.com/.

[107] J. Ens and P. Pasquier, "Mmm: Exploring conditional multi-track music generation with the transformer," *arXiv*, no. arXiv:2008.06048, Aug. 2020, arXiv:2008.06048 [cs]. DOI: 10.48550/arXiv.2008.06048. [Online]. Available: http://arxiv.org/abs/2008.06048.

[108] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th international conference on music information retrieval (ISMIR 2011)*, 2011.

4. Appendix

4.1 Comparing Tokenization lengths

Pretokenized Sequence Lengths - Mean: 47976.146146146144, Median: 43692.0, Std Dev: 23159.918513030065
Tokenized Sequence Lengths - Mean: 14519.587587587588, Median: 13159.0, Std Dev: 7197.064427699848

Applying the BPE tokenizer results in sequence lengths roughly 1/3 of the original sequences. The sample is taken from 1000 random music pieces from the lakh-midi dataset.

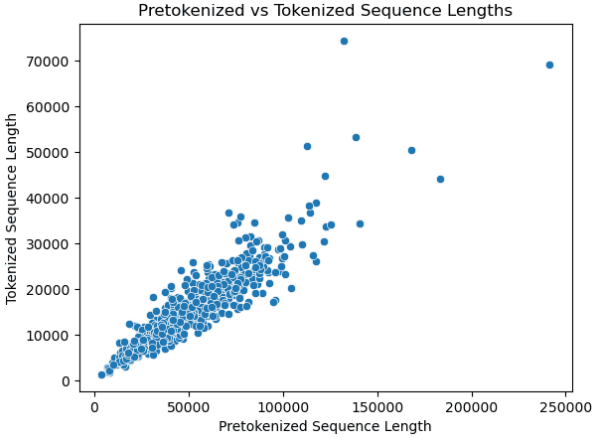

Figure 4.1: Scatterplot of Pre-tokenization vs bpe-tokenization sequence lengths

Name	Architecture	Control	FG	MI	Dataset	Representat
DeepBach (2017) [13]	RNN	Inpainting	Yes	Yes	[99]	Midi-Like
FolkRNN (2015) [14]	RNN	meter, mode	No	No	[100]	REMI-Like
MT (2018) [67]	Transformer	-	No	No	[101] [99]	Midi-Like
MidiNet (2019) [12]	GAN	chords, melody	Yes	Yes	[102]	Midi-Like
Polyfussion 2023 [56]	Diffusion	inpainting),texture	Yes	No	[103]	Piano-Roll
FIGARO 2023 [53]	Transformer	chords, instr, meter, nd	Yes	Yes	[98]	REMI+
MMT 2023 [45]	Transformer	instrumentation	No	Yes	[98],[104]	Midi-Tuple
OMT 2020 [43]	Transformer	chords, tempo	Yes	No	C	REMI
MuseNet 2019 [16]	Transformer	instr, genre	Yes	Yes	[101][105][106]	?
MMM 2020 [107]	Transformer	inpainting, instr, nd	Yes	Yes	[98]	C
Sympack 2024 [39]	Transformer	chords, structure, notes	Yes	Yes	[98] [108], C	
MuseCoco 2023 [51]	Transformer	various	No	No	C	REMI-Like
MBC 2024 [90]	Transformer	chords	Yes	No	[103]	REMI-Like
Museformer 2022 [97]	Transformer	-	No	Yes	[98]	REMI-Like
NTT [44]	Transformer	-	No	Yes	[98] [103] [104]	Compound
FTG [61]	Diffusion	texture, rhythm, chords	Yes	No	[103]	Piano-Roll
Fader Nets	VAE	rhythm, arousal	No	No	[101]	Perf
NDRD [59]	Diffusion	chord, pitch, nd	Yes	No	[101] [103]	Piano-Roll

Table 4.1: Overview of music generation models, their architectures, and control mechanisms. Empty columns indicate missing information on fine-grained control, multitrack capabilities, dataset, and evaluation methods.

4.2 Notes Comparing Symbolic Music Generators and their evaluation methods

- DeepBach [13] - RNN - inpainting. Evaluation Turing
- FolkRNN [14] - RNN - control for meter and mode - Expert Evaluation + Performance Practice
- MusicTransformer [67] - Transformer - Evaluation: Subjective - Tournament Style between different generated and natural music. Objective: Validation NLL
- MidiNet[12] - GAN - Control for Chords/Priming melody. Human (how pleasing, how real, and how interesting)
- Polyfussion [56] - Diffusion Model - supports inpainting, interpolation, melody/accompaniment generation, control for chord progression, texture. Subjective Evaluation Questionnaire for naturalness, creativity, musicality. Objective Control success.
- FIGARO [53] - Transformer Model - bar-wise control for chords, instrumentation, time-signature, note-density, mean-pitch. Evaluation: Perplexity (improvement over NLL for sequences of different length), Discription Fidelity (i.e accuracy in regards binary controls instruments, chords, time-signature). Macro Overlapping Area - comparison of feature histograms. Normalized Mean Root Square Area for note-density. Cosine similarity for chroma (melodic) and groove (rhythmic) feature vectors. Ablation study - effect of turning off controls. Extensive Subjective evaluation: 7569 comparisons by 691 participants - tournament style.
- Multi Track Music Transformer (MMT) [45] - Transformer - Instrument control. Comparison of different tokenisation techniques, REMI+ and Compound Tokens. Compound tokens are more condensed and the generated samples are longer, achieves significant speedups and reduces memory usage (2.6 * MMM, or 3.5 * REMI+). Objective evaluations: Inference time, pitch class entropy, scale consistency, groove consistency, Human evaluation (90 comparisons by 9 participants) on Coherence Richness Arrangement Overall.

  Additional: Analysis of self attention as explanation avenue, which notes are most important.
- REMI pop music transformer - [43] - Transformer - continuation, control local control over chord and tempo
- MuseNet [16] - Transformer -,Instrument control, style control. No evaluation, only showcase.
- MMM [107] - Transformer model - inpainting, instrument control, note-density. Introduce novel representation No Rigorous Evaluation
- SymPAC [39] - Transformer - Control for Chords, structure, instrumentation, single notes. Train with both symbolic and transcribed audio data. Fine grained control. Constrained generation with a Finite State Machine. Comparison of three separate models trained on three datasets. **Evaluation** of controlability with KL-divergence on different controlled features over chords, structure, and individual notes. - KL divergence decreases with dataset size. 800 samples are generated and compared against a validation set of 3000 songs. Subjective evaluation (12 expert participants MIR researchers and music producers) on parameters of Coherence Richness Arrangement Structure
- MuseCoco [51] - Transformer - Control via text for following attributes instrumentation, ambitus, rhythm (intensity and “dancability”), number of bars, time signature, key,

tempo, duration, artist, emotion, genre. All of these controls are global controls, a description is converted into a list of attributes which is then used for generation. Combination of many datasets. Creation of text descriptions from attribute list with data from the dataset, and ChatGPT. Evaluation: Objective - text to attribute list. Subjective evaluation 19 participants with at least basic music knowledge - questions to Musicality, Controlability (adherence of sample to music description), Overal Impression. Comparison with LLM generated music (with no special training for music generation)

- MBD [90] Extension of MuseCoco for time varying chord controls. Counterfactual Loss and Auxiliary task training improve controllability.
- Museformer [97] - Transformer - No controls. Goal improve long-term structure with fine and coarse attention. Captures structure well. Objective Evaluation: Perplexity: prediction accuracy of next token, Similarity Error the error between the similarity distribution of training data and generated music Subjective Evaluation 10 Participants Musicality, Long Term Structure, Short Term Structure. Ablation study - evaluate objective effect of coarse and fine-grained attention. + Case study and detailed look at model.
- NMT [44] - Transformer improve longterm structure and reduce sequence length through compound tokens. Application to both symbolic and audio tokens. Cross attention vs self-attention comparison Evaluation: FAD, CLAP, KL and NLL over audio tokens. NLL over symbolic tokens. Subjective Evaluation Coherence, Richness, Consistency, Overall 29 participants. 8 selected prompts, 4 different continuations with REMI, Compound Word and 2 NMT variations.
- FTG - Fine Grained Texture Control - Diffusion - control over texture, rhythm and chords.
- Fader Nets[52] - VAE - Control over rhythm, arousal, Idea: develop a "fader" representing a high-level abstract feature i.e arousal. Arousal is disentangled using a VAE into lower level features (i.e rhythmic density).

  Evaluation of the influence of latent features is on generated (style transfer) music.: Consistency, Restrictiveness (one latent dimension does not influence other musical features), Linearity (linear change in latent feature - linear change in musical feature) Subjective listening test to indicate success of arousal shift -> 48 participants, evaluate agreement with arousal direction.
- NDRD Symbolic Music Generation with Non-Differentiable Rule Guided Diffusion. Guidance of diffusion sampling with non-differentiable rules.