Creating and Collecting Meaningful Musical Material with Machine Learning

By

Jonathan Gillick

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

in

Information Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:
Professor David Bamman, Chair
Professor Kimiko Ryokai
Professor Carmine-Emanuele Cella

Fall 2022

Abstract

Creating and Collecting Meaningful Musical Material with Machine Learning

By

Jonathan Gillick

Doctor of Philosophy in Information Science

University of California, Berkeley

Professor David Bamman, Chair

This dissertation explores how machine learning and artificial intelligence can be applied within music composition and production. My approach in this research stems from an underlying perspective that these technologies are deeply intertwined with the people who use them or are affected by them: we can't hope to understand one side of the picture without looking at the other. From this vantage point, I explore the following questions: How do we design algorithms, datasets, and models to support the processes of composers, producers, and other creators? How can we design meaningful interactions with these algorithms? And finally, how do music creators and listeners experience interacting with algorithms for creating music in situations when they have reasons to be emotionally invested in the music?

First, I explore new approaches to music creation technology with machine learning, focusing on two musical settings: beat-making and orchestration. I find that creative tools can benefit from incorporating machine learning if we introduce models in specific contexts motivated by well-defined musical goals. For beat-making, I use machine learning to expand the possibilities for *groove* in drum machines by modeling the timing and dynamics of professional drummers, and for orchestration, I use machine learning to predict the timbral characteristics that result when mixing many different instruments together. In both cases, I find that careful data collection and management are key components.

Next, I investigate some of the main technical choices that need to be made when using machine learning in creative musical contexts: data representations and controls for guiding models. Here, I find that allowing users to provide more than one demonstration at a time can allow for more diverse and more precisely controlled model outputs. I also find that using data representations designed to capture musical gestures can provide benefits in settings with limited musical data.

Finally, in the last part of the dissertation, I conduct a series of qualitative studies to investigate the individual experiences of listeners and musicians over the course of interactions with algorithms that generate personalized music samples using machine learning. I find the musical "quality" of algorithmically generated music to be comparatively unimportant to participants; instead, the degree to which music fits with the participants' existing narratives or creative intentions matters more to them. These findings highlight the need to understand the contexts in which algorithms are deployed as well as the artistic choices that listeners may or may not feel comfortable turning over to an automated process when working with emotionally sensitive materials.

Acknowledgements

To the people who supported me along the way: Thank you so much! I wouldn't have made it here without all of you.

Thank you to my advisor David Bamman for teaching me about research, patiently answering lots of naive questions, and showing me how to navigate the strange bubble of Academia. Thanks to Kimiko Ryokai for valuing and making space for my creative side in research and for inspiring me with passion and enthusiasm. Thanks to Carmine Cella for being a positive support, making work fun and fascinating, and for keeping it real with me.

Thank you to all the people at the I School for creating a special community and for caring about the things that matter more than work - community, culture, kindness, justice. I learned so much from all of you and the examples you set. Special thanks to the group that I was fortunate to have around near the beginning - Andrew Chong, Anne Jonas, Daniel Griffin, Elizabeth Resor, Guanghua Chi, Jeremy Gordon, Max Curran, Matt Sims, Nick Merrill, Nitin Kohli, Noura Howell, Richmond Wong, Shazeda Ahmed, and to others keeping me going toward the end - Kent Chang, Lucy Li, Pierre Tchetgen. Thanks to the I School staff and faculty - Inessa Gelfenboym Lee, Catherine Cronquist Browning, Coye Cheshire, and many more - for making this the place that it is.

Thank you to musical collaborators through these years - especially Gavin Bodkin, Aspen Jordan, Nick Woodbury, Max Savage - for inspiring me to keep playing and creating. And thanks to the music teachers who opened my ears to things I needed to hear to to the work in this dissertation - Paul Speidel, Matt Donner, David Earl, Jerome Rossen, Abraham Adzenyah, CK Ladzekpo. These pages may look like a lot of words and not a lot of sound, but it's really about the music.

Thank you to Adam Roberts and the rest of the Magenta team for supporting and investing in me and for giving me a chance to try out my work in front of lots of people.

Thank you to Anna Huang, Hendrik Vincent Koops, Bob Sturm, and the many others who volunteer their time to organize and support events like the AI Song Contest and academic venues for the community working on ML, music, and creativity.

Thank you to Rebecca Fiebrink and Mick Grierson for investing in me for the next chapter of my career at my postdoc and giving me lots to be excited about during the last tough days of trying to finish my dissertation.

Thank you to Bob Keller for starting me on this journey such a long time ago with your kind and generous mentorship at a formative time. You will be missed.

Thank you to my brother Daniel and the rest of my family for paving the way and making me believe that my getting a PhD was something I could imagine doing.

Thanks to Melody for being the best inspiration, partner and human imaginable. And thanks to Kai for giving me both metaphorical and literal kicks in the butt to finish this thing.

Contents

Chapter 1

Introduction

In April 2020, employing hundreds of Graphics Processing Units (GPU's) and tens of thousands of hours of computation time, OpenAI released Jukebox, a giant (by 2020's standards) machine learning model for automatically generating music with a single click of a button [1]. Trained on a catalogue of 600,000 popular music tracks together with metadata scraped from the internet, Jukebox is able to synthesize high resolution audio that mimics the voices, lyrics, performances, and compositions of artists past and present. OpenAI's research paper, along with an accompanying website for browsing thousands of uncanny generated songs featuring ”deepfake” versions of real musicians, sparked a flurry of concern among artists as well as a variety of other stakeholders throughout the music industry.

Within a week of Jukebox's release, hip-hop artist Jay-Z submitted a copyright claim to YouTube through his company Roc Nation, requesting that two videos containing deepfaked audio of Jay-Z be taken down [2]. Though these fake versions of Jay-Z's voice were actually not created with OpenAI's software (anonymous YouTube creator ”Vocal Synthesis” used Google's text-to-speech model Tacotron 2 [3]), Roc Nation's reaction foreshadowed a growing uncertainty and unease within creative industries and creative communities about what the impacts of AI generated content might be.

Systems like Jukebox certainly offer intriguing creative possibilities for music production, composition, sound design, and other musical activities. At the same time, technologies based on machine learning often turn out to be difficult for musicians to use in meaningful ways (e.g. because they create irrelevant results or are cumbersome to interact with). They also can easily end up harming the same communities that they are nominally designed to support (e.g. by exploiting the *ghost work* [4] of artists through the use of music as training data without compensation). Despite impressive progress in machine learning methods in recent years, designing useful and easy-to-use machine learning technology to support *meaningful* creative practices for musicians is a difficult challenge that requires engaging deeply with both the technology itself and the artistic practices, processes, and communities of the people who might use it.

As I write this thesis in 2022, it remains to be seen how music generation technology will be broadly deployed, adopted, or regulated in practice [5]. The effects of new technologies on musical culture can be shaped dramatically by the specific affordances those technologies provide (like sampling on Akai's MPC), along with the cultural practices of niche communities formed by early adopters (like hop-hop artists using the MPC in the late 1980's). For this reason, the current moment in time appears to be a pivotal one: practices that gain traction over the next few years among communities of artists, researchers, and practitioners hold potential to shape the future of both AI's impact on artists and artists' impact on AI.

This dissertation takes some small steps to explore one artist-centric vision for how machine learning might be applied to music. Over the course of the chapters that follow, I ask: How might the same kind of technology used to mimic existing music instead be designed and used to help individuals produce unique music that draws from their individual experiences? Rather than using AI to copy from or create echoes of earlier work, how can we leverage AI to honor these influences, learn from them, and ultimately synthesize them with our own unique experiences to express ourselves through new music and art?

Part 1: Building New Machine Learning Models for Music Creation

The first part of the dissertation presents two attempts to use machine learning to expand the creative possibilities offered by digital production or composition tools. Chapter 2 focuses on making beats, reimagining how drum machines might be designed with more flexible notions of rhythm and groove by modeling the nuances of how professional drummers play the drum set. Chapter 3 applies machine learning to musical orchestration with the aim of assisting composers by predicting the timbral characteristics that might result from mixing many different musical notes and instruments together.

I hope that these two examples can serve as useful references for music technology designers and developers who are interested in applying machine learning to new musical scenarios. These chapters illustrate a range of design, data collection, data representation, and modeling problems that can arise in the process and offer examples of how to approach those problems.

Part 2: Designing for Musical Interaction with Machine Learning

Chapters 4 and 5 dig further into the details of designing and building machine learning models for interactive use, focusing on *embodied* musical concepts like groove and gesture. In Chapter 4, I explore how models for generating drum loops can be designed to facilitate interaction via accepting multiple user-provided demonstrations that describe different characteristics of what a model's output should sound like. Chapter 5 takes a closer look at how low-level choices of data representation can have an impact on the affordances that models can ultimately provide for end users.

I hope that these chapters can provide insight about how modeling decisions might shape the experiences of users and illustrate technical and musical considerations that can play a role in those decisions.

Part 3: Musician and Listener Perspectives on Interactions with AI

In Chapter 6, I look at musical interactions with AI through a series of three user studies with listeners and musicians. While the preceding chapters fit for the most part into machine learning or technically-oriented human-computer interaction research disciplines, Chapter 6 is less concerned with offering technical solutions. Here, I use qualitative design research to develop a better understanding of subjective, emotional experiences as people interact with the kind of technology developed in earlier chapters.

This chapter is intended for both artists and researchers, and I hope that it sheds light on the importance of context and process whenever people use machine learning in the course of music creation.

Chapter 2

Designing for Music Creation with Machine Learning in the Loop: Modeling Groove in Drum Machines[1]

Abstract

This chapter explores models for translating abstract musical ideas (scores, rhythms) into expressive performances using Seq2Seq and recurrent variational Information Bottleneck (VIB) models. Though Seq2Seq models usually require painstakingly aligned corpora, we show that it is possible to adapt an approach from the Generative Adversarial Network (GAN) literature (e.g., Pix2Pix [7] and Vid2Vid [8]) to sequences, creating large volumes of paired data by performing simple transformations and training generative models to plausibly invert these transformations. Music, and drumming in particular, provides a strong test case for this approach because many common transformations (quantization, removing voices) have clear semantics, and models for learning to invert them have real-world applications. Focusing on the case of drum set players, we create and release a new dataset for this purpose, containing over 13 hours of recordings by professional drummers aligned with fine-grained timing and dynamics information. We also explore some of the creative potential of these models, including demonstrating improvements on state-of-the-art methods for Humanization (instantiating a performance from a musical score).

2.1 Introduction

A performance can be viewed as a translation of an idea conceived in the mind to a finished piece on the stage, the screen, or the speakers. The long-standing goal of many creative technologies is to enable users to render realistic, compelling content

[1] The material is this chapter draws upon my previously published work in *Learning to Groove with Inverse Sequence Transformations* at the 2019 International Conference on Machine Learning [6] with co-authors Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman.

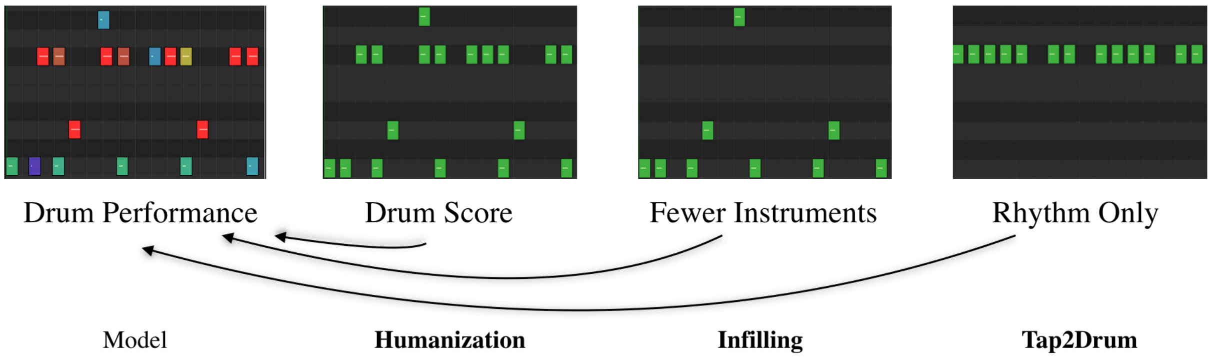


Figure 2.1: Learning inverse sequence transformations for drumming. Moving from left to right, the representations become progressively simpler, first removing expressive timing (small shifts off the grid) and dynamics (color, with higher velocities in red), then removing one of the voices, and then compressing all voices to a single track. We train models to map from each of these deterministically compressed representations back to complete realizations of drum performances. The inverse transformations correspond to Humanization, Infilling, and Tap2Drum respectively, and require progressively easier inputs for an untrained user to create.

that brings an idea to life; in so doing, finding a balance between realism and control is important. This balance has proved difficult to achieve when working with deep generative models, motivating recent work on conditional generation in several modalities including images [9], speech [3], and music [10]. In this work, rather than generating new content conditioned on one of a fixed set of classes like *rock* or *jazz*, we are interested in learning to translate ideas from representations that are more easily expressed (musical abstractions such as scores) into instantiations of those ideas that would otherwise be producible only by those skilled in a particular instrument (performances).

We use the metaphor of translation from idea to finished work as a starting point for our modeling choices, adapting and modifying Seq2Seq models typically used in machine translation [11]. While musical scores and performances can be thought of as different expressions of the same idea, our setting differs from translation in that musical scores are designed to be compressed representations; the additional information needed to create a performance comes from the musician. In this chapter, we set up a data collection environment in which a score can be deterministically extracted from the performance in a manner consistent with the conventions of western music notation, effectively yielding a parallel corpus. Furthermore, though western music notation is well established as one compressed representation for music, our data allows us to explore other representations that are compressed in different ways; we propose and explore two such transformations in this chapter, which we call Infilling and Tap2Drum. Learning to map from these reduced versions of musical sequences to richer ones holds the potential for creative application in both professional and amateur music composition, production, and performance environments.

This chapter focuses specifically on drums; though drums and percussion are essential elements in many styles of modern music, creating expressive, realistic sounding digital drum performances is challenging and time consuming. Humanization functions have been embedded in industry standard music production software for many years, but despite evidence that the current methods used in professional toolkits (randomly jittering note timings and dynamics with Gaussian noise) have little effect on listener preferences [12], machine learning based methods have not yet made their way into many mainstream environments to replace them. We hope that our data, models, and methods for generating and controlling drum performances will continue to drive forward the growing body of work on expressive performance modeling.

In summary, this chapter makes the following contributions:

- We collect a new dataset an order of magnitude larger than the largest previously publicly available, with 13.6 hours of recordings of 10 drummers playing electronic drum kits instrumented with sensors to capture precise performance characteristics in MIDI format. We pair this data with associated metadata including anonymized drummer identifiers, musical style annotations, and tempo, while also capturing and aligning the synthesized audio outputs.
- We present a data representation and a class of models that we call GrooVAE. We use our models to explore the task of Humanization, learning to perform a musical score for drum set, demonstrating improvements over previous methods.
- We introduce, implement, and evaluate two new tasks made possible by our data and model, which we call Drum Infilling and Tap2drum. We argue that these models, along with Humanization, may allow for user control over realistic drum performance generation without expertise playing the drum set.

Code, data, trained models, and audio examples are available online. [2]

2.2 Related Work

A small number of previous studies explore machine learning methods for generating expressive drum performance timing, employing linear regression and K-Nearest Neighbors [13], or Echo State Networks [14]. These studies use data from different musical genres and different drummers, so relative performance between methods is not always clear. In most cases, however, listening tests suggest that qualitative results are promising and can produce better outputs than those created heuristically through a *groove template*[3] [13].

Other expressive performance modeling focuses on piano rather than drums, leveraging data from performances recorded on electronic keyboards or Disklaviers, pianos instrumented with MIDI inputs and outputs [15, 16, 17, 18, 19]. Recent impressive results in generating both MIDI and audio also suggest that given enough data, neural sequence models can realistically generate expressive music. One drawback of the large

[2] https://g.co/magenta/groovae

[3] Groove templates, which are used commonly in music production practice, copy exact timings and velocities of notes from a template sequence.

piano datasets, however, is that they lack gold standard alignments with corresponding musical scores, making tasks like Humanization more challenging.

There are of course many other settings besides music in which learning to translate from abstractions to instantiations can be useful. State-of-the-art methods for speech synthesis [20, 21], and story generation [22] typically use Seq2Seq frameworks. Unlike our case, however, these methods do require paired data, though some recent work attempts to reduce the amount of paired data needed through self-supervised learning [23].

Perhaps most similar to our setting is recent work in the image domain, which has demonstrated the ability of GAN models to translate from simple, potentially user-provided, inputs into photo-realistic outputs [7, 8]. Images and music are similar in that their contents can survive abstraction into simplified versions through lossy transformations like quantization or edge detection while still retaining important semantic details. Images, however, are structured fundamentally differently than musical sequences and tend to benefit from different modeling choices – in particular the use of GANs, which have not been demonstrated to work as well for music as recurrent neural networks.

2.3 Data

Existing work on expressive drum modeling focuses only on small datasets with a limited number of sequences, drummers and genres [13, 24]. Other studies that model expressive performance on different instruments (typically piano) use larger and more diverse datasets [18, 25, 19], but these data lack ground truth alignments between scores and performances; this alignment, which allows use to measure time relative to a metronome, is key to the applications we explore in this chapter. Several companies also sell drum loops played to a metronome by professional drummers, but these commercially produced loops may be edited in post-production to remove human error and variation, and they also contain restrictive licensing agreements that prohibit researchers from sharing their models.

There is currently no available gold standard dataset that is of sufficient size to reasonably train modern neural models and that also contains a precise mapping between notes on a score and notes played by a performer.

Groove MIDI Dataset

To enable new experiments and to encourage comparisons between methods on the same data, we collect a new dataset of drum performances recorded in MIDI format (the industry standard format for symbolic music data) on a Roland TD-11[4] electronic drum kit. MIDI notes (we also refer to them as hits) are each associated with an instrument, a time, and a velocity. Microtimings, (we also call them timing offsets), describe

[4] https://www.roland.com/us/products/td-11/

Figure 2.2: A drummer recording for the Groove MIDI Dataset

how note timings stray from a fixed grid, and velocities (or dynamics) denote how hard notes are struck. Taken together, we refer to microtiming and velocity as *performance characteristics* or *groove*, and the quantized times of the notes define a musical score (also called a pattern or sequence). While some nonpercussive instruments like strings or horns, which allow for continuous changes to a single note, are difficult to represent with MIDI, many styles of drum set playing can be well specified through microtiming and velocity. This dataset, which we refer to as the Groove MIDI Dataset (GMD)[5], has several attributes that distinguish it from existing ones:

- The dataset contains about 13.6 hours, 1,150 MIDI files, and 22,000 measures of drumming.
- Each performance was played along with a metronome set at a specific tempo by the drummer. Since the metronome provides a standard measurement of where the musical beats and subdivisions lie in time, we can deterministically quantize all notes to the nearest musical division, yielding a musical score. Recording to a metronome also allows us to take advantage of the prior structure of music by modeling relative note times (quarter note, eighth note, etc.) so as to free models from the burden of learning the concept of tempo from scratch. The main drawback of the metronome is that we enforce a consistent tempo within each individual performance (though not across performances) so we do not capture the way in which drummers might naturally change tempo as they play.

[5] https://magenta.tensorflow.org/datasets/groove

- The data includes performances by a total of 10 drummers, 5 professionals and 5 amateurs, with more than 80% coming from hired professionals. The professionals were able to improvise in a wide range of styles, resulting in a diverse dataset.

- The drummers were instructed to play a mix of long sequences (several minutes of continuous playing) and short beats and fills.

- Each performance is annotated with a genre (provided by the drummer), tempo, and anonymized drummer ID.

- Most of the performances are in 4/4 time, with a few examples from other time signatures; we use only the files in 4/4 in this work.

- In addition to the MIDI recordings that are the primary source of data for the experiments in this work, we captured the synthesized audio outputs of the drum set and aligned them to within 2ms of the corresponding MIDI files. These aligned audio files may serve as a useful resource for future research in areas like Automatic Drum Transcription.

- A train/validation/test split configuration is provided for easier comparison of model accuracy on various tasks.

Preprocessing

Though the Groove Midi Dataset contains all the information captured by the electronic drum kit, including multiple sensors to detect hits on different parts of each drum, we make several preprocessing choices to simplify our models for this work. First, we map all drum hits to a smaller set of 9 canonical drum categories, following Roberts et al. [26]. These categories represent the most common instruments in standard drum kits: bass drum, snare drum, hi-hats, toms, and cymbals. Table 2.1 displays the choice of MIDI notes to represent the nine essential drum voices for this study, along with the counts of each pitch in the data.

After partitioning recorded sequences into training, development, and test sets, we slide fixed size windows across all full sequences to create drum patterns of fixed length; though we explored models for sequences of up to 16 measures, for consistency we use 2 measure (or 2 *bar*) patterns for all reported experimental evaluations, sliding the window with a hop size of 1 measure. We chose 2 measures for our experiments both because 2 bars is a typical length for drum loops used in music production practice and because these sequences are long enough to contain sufficient variation but short enough to quickly evaluate in listening tests.

As a final step, motivated by the fact that music production software interfaces typically operate at 16th note resolution [27], we take 16th notes as the fundamental timestep of our data. Each drum hit is associated with the closest 16th note metrical position; if multiple hits on the same drum category map to the same timestep, we

Pitch	Roland Mapping	GM Mapping	Drum Category	Count
36	Kick	Bass Drum 1	Bass (36)	88067
38	Snare (Head)	Acoustic Snare	Snare (38)	102787
40	Snare (Rim)	Electric Snare	Snare (38)	22262
37	Snare X-Stick	Side Stick	Snare (38)	9696
48	Tom 1	Hi-Mid Tom	Mid Tom (48)	13145
50	Tom 1 (Rim)	High Tom	High Tom (50)	1561
45	Tom 2	Low Tom	Low Tom (45)	3935
47	Tom 2 (Rim)	Low-Mid Tom	Mid Tom (48)	1322
43	Tom 3 (Head)	High Floor Tom	Low Tom (45)	11260
58	Tom 3 (Rim)	Vibraslap	Low Tom (45)	1003
46	HH Open (Bow)	Open Hi-Hat	Open Hi-Hat (46)	3905
26	HH Open (Edge)	N/A	Open Hi-Hat (46)	10243
42	HH Closed (Bow)	Closed Hi-Hat	Closed Hi-Hat (42)	31691
22	HH Closed (Edge)	N/A	Closed Hi-Hat (42)	34764
44	HH Pedal	Pedal Hi-Hat	Closed Hi-Hat (42)	52343
49	Crash 1 (Bow)	Crash Cymbal 1	Crash Cymbal (49)	720
55	Crash 1 (Edge)	Splash Cymbal	Crash Cymbal (49)	5567
57	Crash 2 (Bow)	Crash Cymbal 2	Crash Cymbal (49)	1832
52	Crash 2 (Edge)	Chinese Cymbal	Crash Cymbal (49)	1046
51	Ride (Bow)	Ride Cymbal 1	Ride Cymbal (51)	43847
59	Ride (Edge)	Ride Cymbal 2	Ride Cymbal (51)	2220
53	Ride (Bell)	Ride Bell	Ride Cymbal (51)	5567

Table 2.1: List of Drum Categories

keep the loudest one. Although this preprocessing step forces us to discard some of the subtle details of drum rolls that can be played on a single drum faster than 16th notes, we found that perceptually, much of the expressiveness in drumming can be conveyed at this resolution. Moreover, after experimenting both with finer resolutions (32nd or 64th notes) and data representations that count time in absolute time (milliseconds) rather than relative time (as in Simon et al. [25]), we found that the gains in modeling yielded by this constraint were more important than the details lost. One potential path forward in future work might be to supplement our data representation with an explicit token for a drum roll.

Data Represention

After preprocessing, our data points are of fixed length: each sequence has T timesteps (one per 16th note) and M instruments per timestep. The full representation consists of the below three $T \times M$ matrices, with values $T = 32$ and $M = 9$ for all reported experiments.

Hits. To represent the presence or absence of drum onsets, or hits, in a sequence, we define a binary-valued matrix H, which contains all the information in a basic drum score. A column of H contains the drum score for one of the nine instruments in the drum set, and a row of H contains the drum score for all nine instruments at a single timestep.

Offsets. A continuous-valued matrix O stores the timing offsets, taking values in [-0.5, 0.5) that indicate how far and in which direction each note's timing lies relative to the nearest 16th note. Drum hits may fall at most halfway between their notated position in time and an adjacent position. We can examine O to compute statistics on microtiming: positive values indicate playing behind the beat (late); negative values demonstrate playing ahead (early).

Modeling continuous as opposed to discrete values for offsets allows us to take advantage of the fact that timing appears to be approximately normally distributed at any given metrical position (as shown in Figure 2.3); intuitively, models should be penalized more for predictions that are further from the ground truth. We experimented with various continuous and discrete representations including logistic mixtures [28], thermometer encodings [29], and label smoothing [30], but we found that modeling timing offsets and velocity as single Gaussian distributions (conditional on the LSTM state) produced by far the most perceptually realistic results.

Velocities. Another continuous-valued matrix V stores the velocity information (how hard drums are struck). We convert velocity values from the MIDI domain (integers from 0-127) to real numbers in [0,1].

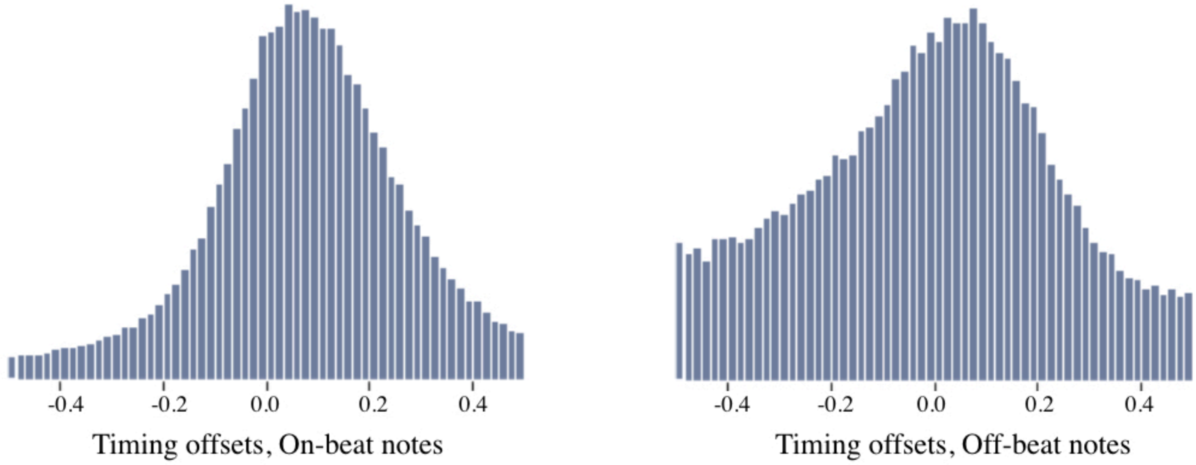


Figure 2.3: Distribution of timing offsets for notes in the training set. On-beat notes (landing on an eighth note), shown on the left, are more often played late, whereas off-beat notes (not landing on an eighth note), on the right, are more often played early.

2.4 Modeling Objectives

We focus our experiments and analysis on three particular applications of expressive performance modeling. For audio examples of additional tasks such as unconditional sampling, interpolation, and style transfer, see the online supplement[6].

Humanization. Our first objective is to generate, given a 16th-note-quantized drum pattern with no microtiming or velocity information (i.e., a drum score), a MIDI performance of that score that mimics how a professional drummer might play it. Because this task has an existing body of work, we focus most of our experiments and evaluations on this task.

Infilling. We introduce a second task of interest within the same contexts as Humanization that we call Drum Infilling. The objective here is to complete or modify a drum beat by generating or replacing the part for a desired instrument. We define an instrument as any one of the 9 categories of drums and train models that learn to add this instrument to a performance that lacks it. For brevity, we choose a single drum category (hi-hat) as a basis for our evaluations. Infilling provides a case for examining computer assisted composition, allowing a composer to sketch parts for some pieces of the drum kit and then receive suggestions for the remaining parts. Previous work explores Infilling in the context of 4-part Bach compositions [31] and in piano performance [32]; we look at the task for the first time in the context of drums.

[6] http://goo.gl/magenta/groovae-examples

Tap2Drum. In this last task, we explore our models' ability to generate a performance given an even further compressed musical representation. While western musical scores usually denote the exact notes to be played but lack precise timing specifications, we propose a new representation that captures precise timing but does not specify exactly which notes to play. In this setting, which we call Tap2Drum, we give our model note offset information indicating the microtiming, but we do not specify the drum categories or velocities as inputs, leaving the decision of which instrument to hit and how hard to the model. Because almost anyone can tap a rhythm regardless of their level of musical background or training, this input modality may be more accessible than musical notation for those who would like to express their own musical ideas on a drum set but lack the skills of a drummer.

2.5 Models

We compare several models for Humanization, selecting the best performing one for our experiments with Infilling and Tap2Drum.

Baselines

For our baseline models, we focus on models from the literature that have been used before for Humanization in the context of drum performances aligned to a metronome.

Quantized

As a simple baseline, we set all offsets to 0 and velocities to the mean value in the training set.

Linear Regression

For this baseline, we regress H against V and O, predicting each element of V and O as a linear combination of the inputs H.

K-Nearest Neighbors

Wright and Berdahl [13] report strong results in using K-Nearest Neighbors to predict microtiming in Brazilian percussion. They define a hand-crafted distance measurement between notes, retrieve the K notes in the training set nearest to a given note in a test sequence, and then take the mean timing offset of those notes. Their definition of nearest notes, however, requires that the same sequence appear in both training and test sets. Since our test set emphasizes unseen sequences, we adapt the method as follows: first we retrieve the K nearest *sequences*, measuring distance $D_{i,j}$ by counting the number of notes in common between a test sequence x_i and each training sequence

x_j, which can be computed easily through the Hadamard product of their respective binary matrices, H_i and H_j:

$$D_{i,j} = \sum H_i \circ H_j \tag{2.1}$$

Given the closest K sequences $[S_1, \ldots, S_K]$, we then compute predicted velocities $\hat{V}$ and offsets $\hat{O}$ by taking the element-wise means of the corresponding V and O matrices:

$$\hat{V} = \frac{1}{K} \sum_k V_k \tag{2.2}$$

$$\hat{O} = \frac{1}{K} \sum_k O_k \tag{2.3}$$

When reconstructing a MIDI sequence, we ignore the entries of $\hat{V}$ and $\hat{O}$ for which the corresponding entry of H is 0.

Choosing $K = 1$ is equivalent to selecting the most similar sequence as a groove template, and choosing K to be the cardinality of the training set yields a single groove template that summarizes the average performance characteristics of the entire set. Through a grid search on the development set, we found that setting $K = 20$ performed best, close to the reported $K = 26$ from Wright and Berdahl [13].

Proposed Models

MLP

To train multilayer perceptron (MLP) neural networks for Humanization, we concatenate the matrices H, V, and O to form a target matrix $y \in R^{T \times (M*3)}$. We pass H into the model as inputs, training the network to minimize the squared error between y and predictions $\hat{y}$. For the MLP, we use a single hidden layer of size 256 and ReLU nonlinearities. We train all our neural models with Tensorflow [33] and the Adam optimizer [34].

Seq2Seq

Sequence to sequence models [11] encode inputs into a single latent vector, typically with a recurrent neural network, before autoregressively decoding into the output space. For this architecture, we process the drum patterns over $T = 32$ timesteps, encoding a drum score to a vector z with a bidirectional LSTM and decoding into a performance with a 2-layer LSTM.

Encoder The encoder is based on the bidirectional LSTM architecture used in Roberts et al. [26], though we change the LSTM layer dimensions from 2048 to 512 and the dimension of z from 512 to 256. At each timestep t, we pass a vector h_t, which is row t of H, to the encoder, representing which drums were hit at that timestep; velocities and

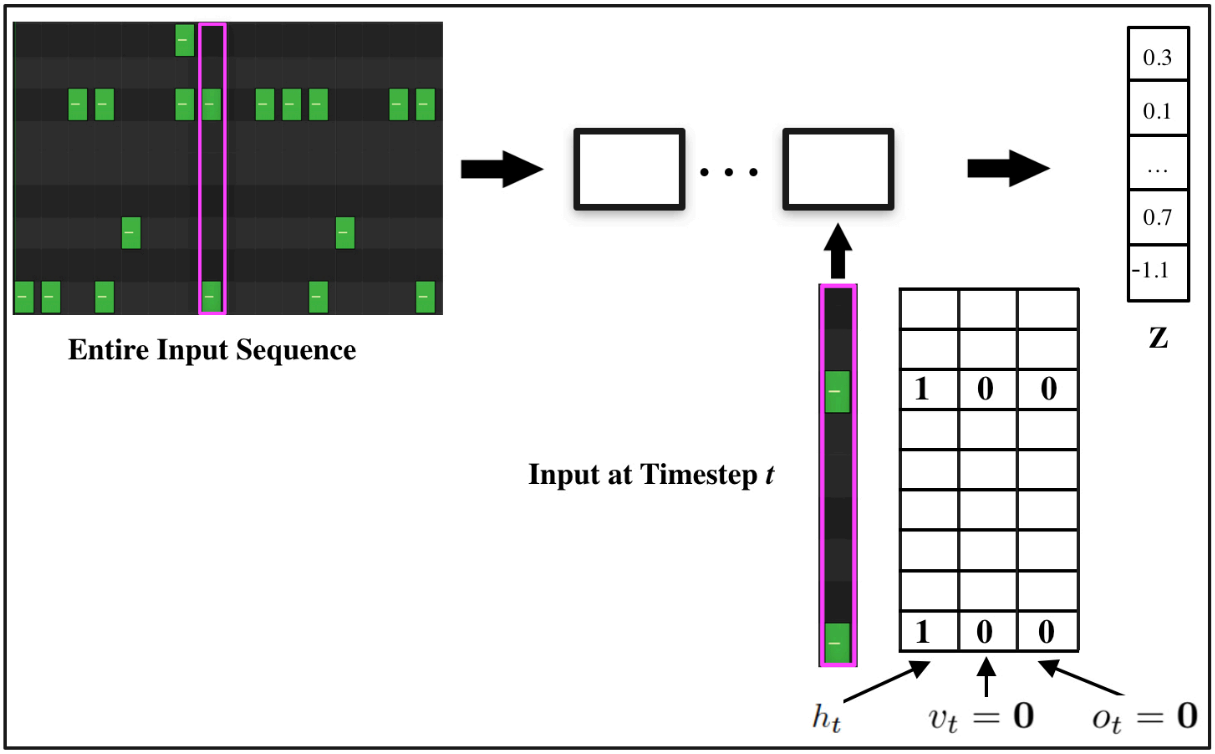


Figure 2.4: The forward direction of our encoder architecture for the Seq2Seq Humanization model. Input sequences are visualized as piano rolls, with drum categories on the vertical axis and time on the horizontal axis. LSTM inputs are shown for a single timestep t. Instruments with no drum hits at time t are shown as blank, although for implementation we fill these blank cells with 0's. Note that no velocity or timing offset information is passed to the encoder.

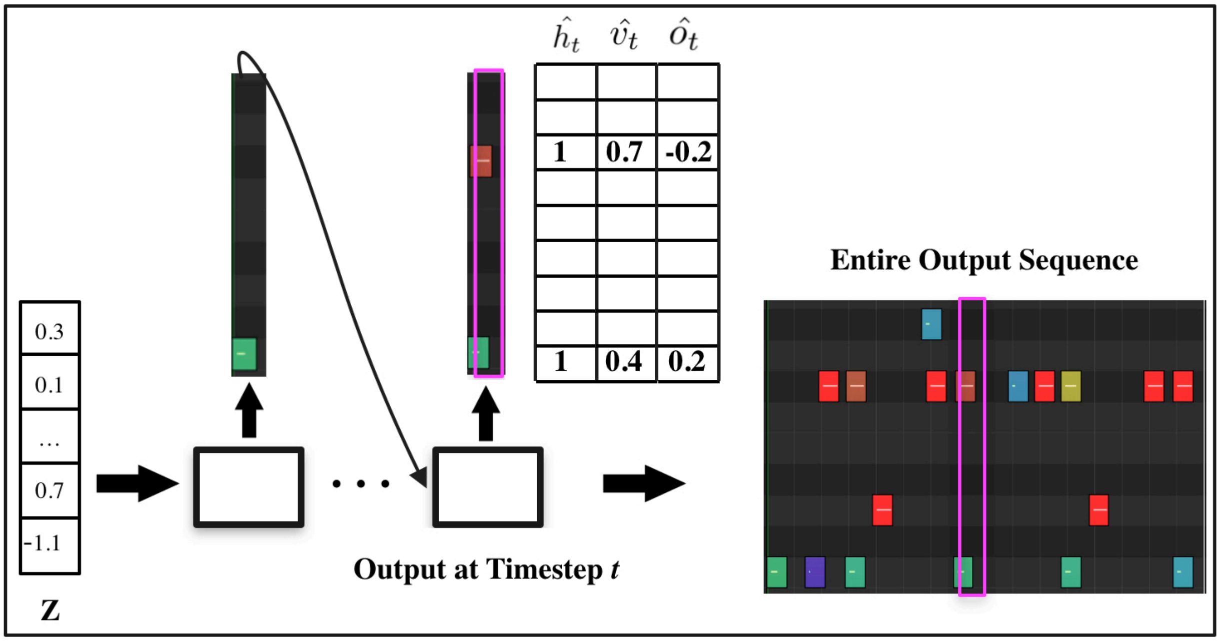


Figure 2.5: Decoder architecture for Seq2Seq Humanization model. The decoder generates outputs for drum hits, velocities and timing offsets. Velocity is visualized in color, and output notes appear slightly earlier than the grid lines, indicating negative offsets.

timing offsets are not passed in. As shown in Figure 2.1, we keep the same architecture for Infilling and Tap2Drum, only modifying the inputs to switch tasks. Figure 2.4 demonstrates one step of the forward direction of the encoder.

Decoder We use a 2-layer LSTM of dimension 256 for our decoder, which we train to jointly model H, V, and O. Unlike Roberts et al. [26], however, we split the decoder outputs at each timestep t into 3 components, applying a softmax nonlinearity to the first component to obtain a vector of predicted hits $\hat{h_t}$, sigmoid to the second component to get velocities $\hat{v_t}$, and tanh to the third, yielding timing offsets $\hat{o_t}$. These vectors are compared respectively with h_t, v_t, and o_t, the corresponding rows of H, V, and O, and finally summed to compute the primary loss for this timestep L_t:

$$L_t = CrossEntropy(h_t, \hat{h_t}) + (v_t - \hat{v_t})^2 + (o_t - \hat{o_t})^2 \tag{2.4}$$

We train the model end to end with teacher forcing.

Groove Transfer

We experiment with one more model that we call Groove Transfer. This architecture is identical to our Seq2Seq model except that at each timestep t we concatenate h_t, the vector for the hits at time t, to the decoder LSTM inputs using the conditioning procedure of Simon et al. [10]. By allowing the decoder to learn to copy h_t directly to its outputs, we incentivize this encoder to ignore H and only learn a useful representation for generating V and O. The main benefit of this architecture over Seq2Seq is that the modification allows us to disentangle the performance characteristics (the groove) of a sequence S_1 from the score H_1, capturing the performance details in the groove embedding z_1. We can then pass z_1 to the decoder along with the content H_2 of another sequence S_2 to do style transfer for drum performances. Audio examples of Groove Transfer can be found in the supplementary materials.

We also apply Groove Transfer to Humanization as follows: given a score H_2, we embed the closest $k = 3$ sequences in the training set as defined by the distance metric in Section 2.5, store the mean of the k embeddings in a vector z_k, and then transfer the groove vector z_k to H_2.

Variational Information Bottleneck

Our test data, while disjoint from the training data, comes from the same set of drummers, and its distribution is meant to be similar. In the real world, however, we would like to be able to trade off between realism and control; when faced with a very unlikely drum sequence, such as one quickly sketched in a music production software interface, we may want to choose a model that constrains its output to be close to the realistic examples in the training set, potentially at the cost of changing some of the input notes. To this end, we add a variational loss term to both Seq2Seq and Groove Transfer, turning the models into a Variational Information Bottleneck (VIB) [35] and training the embeddings z to lie close to a prior (multivariate normal) distribution. Following Roberts et al. [26], we train by maximizing a modified Evidence Lower Bound (ELBO) using the hyperparameter $\beta = 0.2$. We report our quantitative metrics both with and without the VIB.

2.6 Results

Listening Tests

As is the case with many generative models, especially those designed for creative applications, we are most interested in the perceptual quality of model outputs; for this reason, we also highly encourage the reader to listen to the audio examples in the supplementary materials. In our setting, high quality model outputs should sound like real drum performances. We examine our models through multiple head-to-head listening tests conducted on the Amazon Mechanical Turk platform.

Humanization: Comparison with baseline. For this experiment, we compare the Humanization model that we judged produced the best subjective outputs (Seq2Seq with VIB), with the best baseline model (KNN). We randomly selected 32 2-measure sequences from the test set, removing all microtiming and velocity information, and then generated new performances of all 32 sequences using both Humanization models. We presented participants with random pairs of clips, one of which was generated

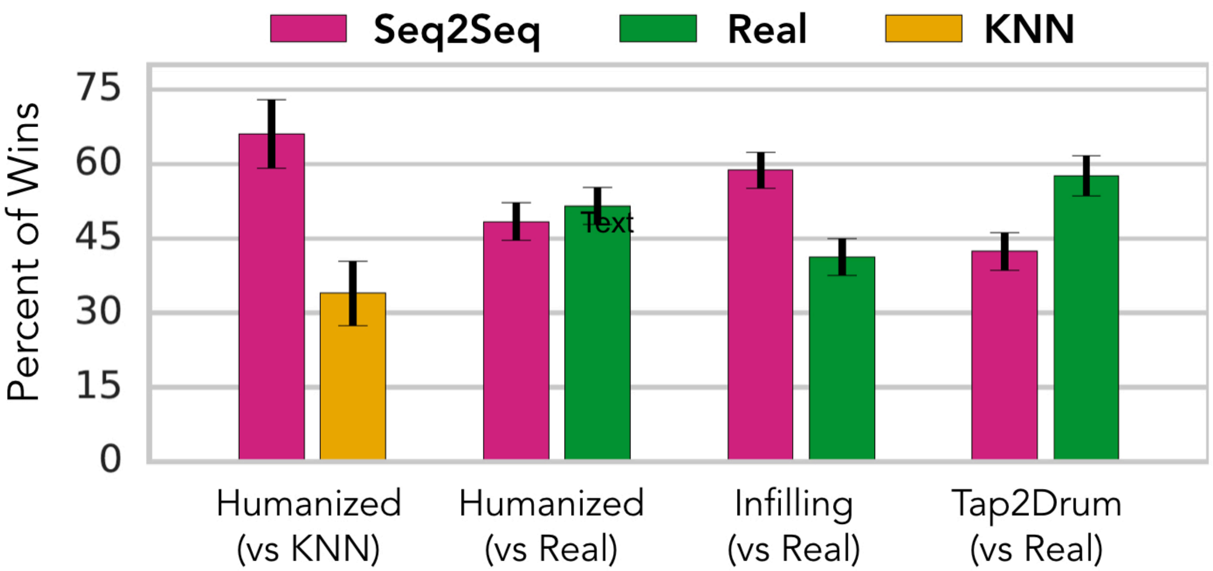


Figure 2.6: Results of head-to-head listening tests for different tasks and baselines, with 95% confidence interval bands. The experiments included 56, 188, 189, and 177 comparisons, respectively.

by each model, asking them to judge which clip sounds more like a human drummer. The Seq2Seq model significantly outperformed the baseline as can be seen in the first column of Figure 2.6.

Comparison with real sequences from the test set. Perhaps a more compelling test of the real-world viability of our models is to ask listeners to compare generated outputs with clips of real drum performances; if the model outputs are competitive, this suggests that the generated drums are perceptually comparable with real performances. We structured this test in the same way as the baseline comparison, asking listeners which sequence sounds more like a human drummer; in this case each pair contains one real clip from the test set and one generated clip. As noted in Section 2.3, because our models do not generate drum rolls faster than 16th notes, we compared against the preprocessed versions of test set clips (which also do not have faster drum rolls) to ensure fair comparison. Figure 2.6 summarizes the results of this test for each of our tasks (Humanization, Infilling, and Tap2Drum), showing the generated outputs from our Seq2Seq models are competitive with real data.

Quantitative Metrics

Though it is difficult to judge these generative models with simple quantitative metrics, we report several quantitative evaluations for comparison, summarizing results in Table 2.2, along with 95% bootstrap confidence intervals.

Model	MAE (ms)	MSE (16th note)	Timing KL	Velocity KL
Baseline	22.6 [22.45–22.72]	0.041 [0.041–0.042]	N/A	N/A
Linear	19.77 [19.63–19.88]	0.033 [0.033–0.034]	4.79 [4.68–4.88]	1.70 [1.66–1.74]
KNN	22.34 [22.19–22.45]	0.043 [0.042–0.0438]	1.10 [1.07–1.12]	0.53 [0.51–0.56]
MLP	19.25 [19.13–19.40]	0.032 [0.031–0.032]	7.62 [7.44–7.80]	2.22 [2.16–2.29]
Seq2Seq	18.80 [18.67–18.90]	0.032 [0.031–0.032]	0.31 [0.31–0.33]	**0.08 [0.08–0.09]**
Seq2Seq+	**18.47 [18.37–18.60]**	**0.028 [0.028–0.029]**	2.80 [2.72–2.86]	0.22 [0.21–0.23]
Transfer	25.04 [24.82–25.28]	0.052 [0.051–0.053]	**0.24 [0.23–0.25]**	0.12 [0.12–0.13]
Transfer+	24.49 [24.25–24.72]	0.051 [0.049–0.052]	0.27 [0.26–0.28]	0.20 [0.19–0.20]

Table 2.2: Metrics for different Humanization models, with 95% bootstrap confidence intervals. Seq2Seq+ and Transfer+ refer to the Seq2Seq model and the Groove Transfer model with the Variational Information Bottleneck included. Seq2Seq and Transfer refer to the same models without the Variational Information Bottleneck.

Timing MAE. We report mean absolute error in milliseconds, which is useful for interpreting results in the context of studies on Auditory Temporal Resolution, a measure of the minimum amount of time required for the human ear to perceive a change in sound. Studies show that temporal resolution depends on the frequency, loudness, and envelope of the sound as well as on the listener and type of recognition test (e.g., noise or pitch recognition) [36, 37]. On tests for which the ear is more sensitive, such as the Gap-in-Noise test, mean values can be as low as 2ms, while for pitched tests like Pure Tone Discrimination, values can be 20ms or more [38]. Most likely, the resolution at which the ear can perceive differences in drum set microtiming lies somewhere in between.

Timing MSE. Following Wright and Berdahl [13], for another perspective on timing offsets, we look at mean squared error relative to tempo, here using fractions of a 16th note as units. Since beats are further apart at slower tempos, this metric weights errors equally across all tempos.

Velocity KL / Timing KL. One drawback of the above metrics, which are aggregated on a per-note basis, is that they do not account for the possibility of mode collapse or blurring when comparing methods [39]. The effects of blurring seem to be particularly severe for velocity metrics; instead of averaging velocity errors across all notes, previous work computes similarity between the distributions of real and generated data [40, 41]. We adopt this approach, first predicting velocities and offsets for the entire test set and then comparing these with ground truth distributions. For these metrics, we aggregate all predicted notes into four groups based on which 16th note position they align with. We calculate the means and standard deviations for each group of notes, compute the KL Divergence between predicted and ground truth distributions based on those means and standard deviations, and then take the average KL Divergence across the four groups. These distribution based metrics should not be treated as a gold standard either, but they do tend to penalize severe instances of blurring or mode collapse, as can be seen with the Linear and MLP models.

2.7 Analysis

Comparisons with KNN baseline

Based on the results of the listening tests shown in Figure 2.6, Seq2Seq models clearly offer a powerful method for generating expressive drum performances. The listener preference for Humanization using Seq2Seq over KNN is substantial, and moreover, these survey participants were not specifically chosen from a pool of expert musicians or drummers; that this pool of listeners was able to so clearly identify the Seq2Seq models as more realistic than the baseline seems to indicate that the model captures important nuances that make drumming realistic and expressive.

Comparing Humanization to real data

The survey results indicate that, at least for our population of listeners, drum performances generated through Seq2Seq Humanization are difficult to distinguish from real data; statistically, the results show no significant difference.

Comparing Infilling to real data

Perhaps counter-intuitively, a significant fraction of listeners in this experiment (nearly 60%) identified the generated outputs as sounding *more* human than the real data. One potential explanation for this result is that among our test data, some sequences sound subjectively better than others. A small fraction of the recordings are from amateur drummers, who sometimes make mistakes or play at a lower level. In replacing the original hi-hat parts, the Infilling model in effect resamples from the data distribution and may generate better sounding, more likely parts. This result suggests a potential use for the model as a corrective tool that works by resampling parts of an input that have noise or imperfections.

Comparing Tap2Drum to real data

Figure 2.6 demonstrates the slight preference of listeners for the real data over performances generated by Tap2Drum (about 56%). This difference is significant but comparatively small relative to the difference between Seq2Seq and KNN Humanization, indicating that Tap2Drum may be a viable way of controlling expressive performances in practice. More work is needed to better understand how much control this model offers and how people interact with the model in different contexts; qualitative research with musicians and music producers offers one path forward.

Groove Transfer

Evaluating Groove Transfer is challenging in the absence of existing methods for comparison; nonetheless, we believe that this particular version of style transfer yields subjectively interesting outputs and merits further investigation both in terms of its architecture and its potential for creative application in the future.

Quantitative Results

As might have been expected, the Seq2Seq models achieve the best results on the timing MAE and MSE metrics, while also outperforming the baselines on the distribution-based metrics. The Groove Transfer models, in exchange for the added control given by the ability to perform a beat in the style of any other beat, sacrifice some accuracy on the Humanization task compared to Seq2Seq, as can be seen by the higher MAE error.

2.8 Conclusions

In this chapter, we demonstrate that learning inverse sequence transformations can be a powerful tool for creative manipulation of sequences. We present the Groove MIDI Dataset, new methods for generating expressive drum performances, and quantitative and qualitative results demonstrating state-of-the-art performance on Humanization.

We also explore new applications, such as Tap2Drum, which may enable novices to easily generate detailed drum performances. Our results raise the possibility of learning other creative inverse transformations for sequential data such as text and audio. We hope this line of research will ultimately lead a variety of interesting creative applications, just as similar GAN-based techniques have done for images and video.

Acknowledgements

Thank you to our drummers for their hard work and talent, without whom this work would not have been possible: Dillon Vado, Jonathan Fishman, Michaelle Goerlitz, Nick Woodbury, Randy Schwartz, Jon Gillick, Mikey Steczo, Sam Berman, and Sam Hancock. Thank you also to the anonymous reviewers at ICML and to Anna Huang, Chris Donahue, Colin Raffel, Curtis "Fjord" Hawthorne, David Ha, Ian Simon, Mike Tyka, Monica Dinculescu, and Yingtao Tian for valuable advice and feedback.

Chapter 3

Expanding the Scope of Music Creation Tools using Machine Learning: Target-Based Orchestration[1]

Abstract

This chapter explores how machine learning can be used to expand the scope of music creation tools, focusing here on the context of computer-assisted musical orchestration. Target-based assisted orchestration can be thought of as the process of searching for optimal combinations of sounds to match a target sound, given a database of samples, a similarity metric, and a set of constraints. A typical solution to this problem is a proposed orchestral score where candidates are ranked by similarity in some feature space between the target sound and the mixture of audio samples in the database corresponding to the notes in the score; in the orchestral setting, valid scores may contain dozens of instruments sounding simultaneously.

Generally, target-based assisted orchestration systems consist of a combinatorial optimization algorithm and a constraint solver that are jointly optimized to find valid solutions. A key step in the optimization involves generating a large number of combinations of sounds from the database and then comparing the features of each mixture of sounds with the target sound. Because of the high computational cost required to synthesize a new audio file and then compute features for every combination of sounds, in practice, existing systems instead estimate the features of each new mixture using precomputed features of the individual source files making up the combination. Currently, state-of-the-art systems use a simple linear combination to make these predictions, even if the features in use are not themselves linear.

In this chapter, we explore neural network models for estimating the features of a mixture of sounds from the features of the component sounds, finding that standard

[1] The material is this chapter draws upon my previously published work in *Estimating Unobserved Audio Features for Target-Based Orchestration* at the 2019 International Society for Music Information Retrieval Conference [42] with co-authors Carmine-Emanuele Cella and David Bamman.

features can be estimated with accuracy significantly better than that of the methods currently used in assisted orchestration systems. We present quantitative comparisons and discuss the implications of our findings for target-based orchestration problems.

3.1 Introduction

In many music information retrieval and signal processing contexts, we are required to reason about signals that are themselves the sum of multiple sources. Whether the summing comes from instruments in a multi-track recording, voices in a group conversation, or simply from noise in the signal, we generally need to consider the full set of sources that make up an audio signal.

Much work in MIR deals with pulling apart the sources in a signal, either in the most straightforward sense via source separation [43, 44], or through any of a number of classification tasks such as tagging [45, 46], instrument recognition [47, 48], or automatic transcription [49, 50]. A separate body of work deals with the inverse problem, that of putting sources together: work in applications like assisted orchestration [51] and automatic mixing [52, 53] aims to guide people through the task of combining signals together with the help of a machine in the loop.

In the cases of both separation and combination, tasks can be solved presumably because the source components and the summed signal are sufficiently correlated; the more correlated a source is with the mixture, the easier it is to identify, and as more signals are summed together, correlations between the combination and any single source tend to diminish. In a computational setting, these correlations are of course measured through a set of features of the signals, whether they be hand-engineered features like FFT and MFCC, or modern features learned by neural networks.

There are some cases, however, in which we can observe the source signals of a mixture, but it is impractical or impossible to actually compute the features of the signal in question; these are the cases that we investigate in this chapter. Broadly speaking, there are two primary settings in which we may be unable to observe the features of audio signals:

1. We do not have access to the signals.

2. Computing the features for *all* relevant signals is computationally expensive.

The first setting is commonly encountered in MIR, in which, as with many fields centered around media that may be under copyright or other protections, it is quite common for researchers to have access to pre-computed features but not to raw data itself. For example, the audio files of the million songs that comprise the Million Song Dataset [54], which serves as a benchmark for many common MIR tasks, cannot be legally distributed. Instead the data contains common audio features like MFCC, chroma, note onsets, and spectral centroids. Though these kinds of dataset are attractive because of their size and scope, they have been of limited use as source material for constructing additional audio mixtures. As semi-supervised and self-supervised approaches to machine learning have become more competitive with fully supervised systems, large datasets even of weakly labeled source material are becoming more useful for research in areas like source separation [55, 56]; estimating the features of mixtures might be one path towards making use of this data in new contexts.

The second setting in which we cannot observe audio features, which is the focus of this chapter, is the case where the computational cost of calculating an exponential number of audio mixtures is prohibitively high. This computational bottleneck often arises in the aforementioned body of work that attempts to automatically combine signals together during the course of tasks like target-based orchestration. In this context, learning algorithms need to explore a combinatorial space of potential solution sets, making it infeasible to compute the real features of all possible mixtures of signals. Moreover, methods for narrowing down this set of possible solutions, such as reinforcement learning algorithms, are generally iterative, requiring online evaluation of a reward function before the next set of candidates can be explored. Because these methods both have a large solution space and need to be evaluated iteratively, features must be computed *on the fly*, making fast feature computation, or accurate estimation, a necessity.

In this chapter, we take steps to explore the potential of machine learning models for predicting audio features of a mixture of sounds that we are unable to observe, focusing on the task of target-based assisted orchestration [51, 57, 58]. Concretely, we consider models of the following form: given a feature function f and M individual signals $S_1, \ldots, S_M$, we learn mappings from input features $f(S_1), \ldots, f(S_M)$ to the true feature of the mixture $f(S_1 + \ldots + S_M)$.

In experiments, we examine one standard feature that is known to typically behave linearly when summed (FFT magnitude spectra) and one feature that is less well suited to linear approximation (MFFC coefficients), investigate the ability of different models to predict each feature across a varying number of mixtures ranging from 2 notes to 30, and discuss the implications of our findings for target-based assisted orchestration as well as for the broader range of scenarios in which real audio features cannot be observed.

Code to reproduce our results can be found at https://github.com/jrgillick/audio-feature-forecasting.

3.2 Target-Based Assisted Orchestration

Musical orchestration, and in many cases, music production, consists largely of choosing combinations of sounds, instruments, and timbres that support the narrative of a piece of music. Strong orchestration can bring a composition to life by emphasizing, clarifying, or perhaps questioning the elements of the music, and through this process, orchestration can often be a difference-maker to critical or commercial success [59, 60].

Different musical styles and composition environments have different constraints (for example, scores for live performance should only require the sounds of the instruments in the group, whereas the sounds available for use on a recording are only limited by their stylistic relevance), but fundamentally, finding the right set of sounds is important regardless of the context. For composers and producers, employing MIR systems during the orchestration process holds the potential to help spark inspiration, solve challenging problems, or save time.

Though the orchestral setting has been explored extensively in previous work, assisted orchestration methods hold the potential for application in other styles. For example, *layering* drum samples is common practice in music production, and MIR-based tools for drum sample search are beginning to make their way into professional toolkits[2]; existing methods for query-based drum sample retrieval [61] could be extended to consider mixtures of drum samples.

3.3 Related Work

Existing systems for target-based assisted orchestration compute spectral similarities using standard spectral features [51] or perceptual descriptors [57], along with evolutionary methods for exploring the solution space.

Most relevant to our experimental setting is the implementation of the publicly available state-of-the art Orchidea system [62], which is based in part on a study conducted in [63] on predicting timbral features of combined sounds. This study found that for 3 features (Spectral Centroid, Spectral Spread, and Main Resolved Partials), and for mixtures of up to 4 sounds, predicting the features of the mixture by a linear combination of the source features both achieved a low error and did not vary as a function of the number of mixture components.

Since computing a linear combination has very low computational cost, this finding enables real-time estimation of thousands of candidate mixtures for use in online reinforcement learning, making tools like Orchidea practical for real-world use. The effects on the features of mixing many more components, along with the behavior of higher-dimensional and richer features, however, have not yet been investigated.

3.4 Experiments

Data

For our experiments, we use the OrchDB dataset of individually recorded musical notes from a variety of orchestral instruments. OrchDB is a streamlined subset of the Studio Online (SOL), dataset that has been optimized for assisted orchestration [64]. Collected as part of the Studio Online project at IRCAM, the full SOL data set contains over 117,000 instrument samples, including extended techniques and contemporary playing styles. OrchDB, which contains a curated subset of these samples, has been used for assisted orchestration since 2008 [65]; the contents of the data are summarized below:

[2] https://www.xlnaudio.com/products/xo

- OrchDB contains about 20,000 notes with lengths ranging from about 1 second to 30 seconds.
- Instruments include bassoon, clarinet, flute, horn, oboe, saxophone, strings, trombone, trumpet, and tuba.
- Approximately 30 different playing styles are represented in OrchDB, such as ordinario, pizzicato, pizzicato Bartók, aeolian, Flatterzung, col legno battuto; brass instrument samples include a variety of different types of mutes.
- Notes across the pitch range are included, along with a range of dynamics from *ppp* to *fff*, including sforzato and crescendo.

Mixtures of Notes

To train and evaluate models for feature estimation, we partition the dataset for training, development, and testing, choose 6 different numbers of mixture components M between 2 and 30, and then for each M, we synthesize a dataset of new audio files by adding together the waveforms of M randomly chosen notes. Finally, we divide the summed signals by M to keep the amplitudes of the mixture in the same range as those of the source files.

For each value of M, we synthesize 7500 new audio mixtures for training, 2000 for development, and 2000 for testing, creating these new mixtures after partitioning our data so that no source file that appears in the training set can be chosen as part of a mixture in the test set. After synthesizing the mixtures, we compute and store the real FFT and MFCC features for every mixture for use in training and evaluating our models.

Predicting Unobserved Features

With this data in hand, we explore several models for predicting the features of a mixture of audio signals given the features of the individual signals. In all experiments, given a feature function f and M individual signals $S_1, \ldots, S_M$, each model is trained to learn a mapping from input features $f(S_1), \ldots, f(S_M)$ to the true feature of the mixture $f(S_1 + \ldots + S_M)$.

3.5 Models

Features

For our modeling experiments, we choose two standard features: 1024-dimensional FFT magnitude spectra and 19-dimensional MFCC coefficients (we discard the first of 20 MFCC coefficients). Our choice of features is meant to capture the most common MIR settings, so we use the default FFT and MFCC dimensions specified in the

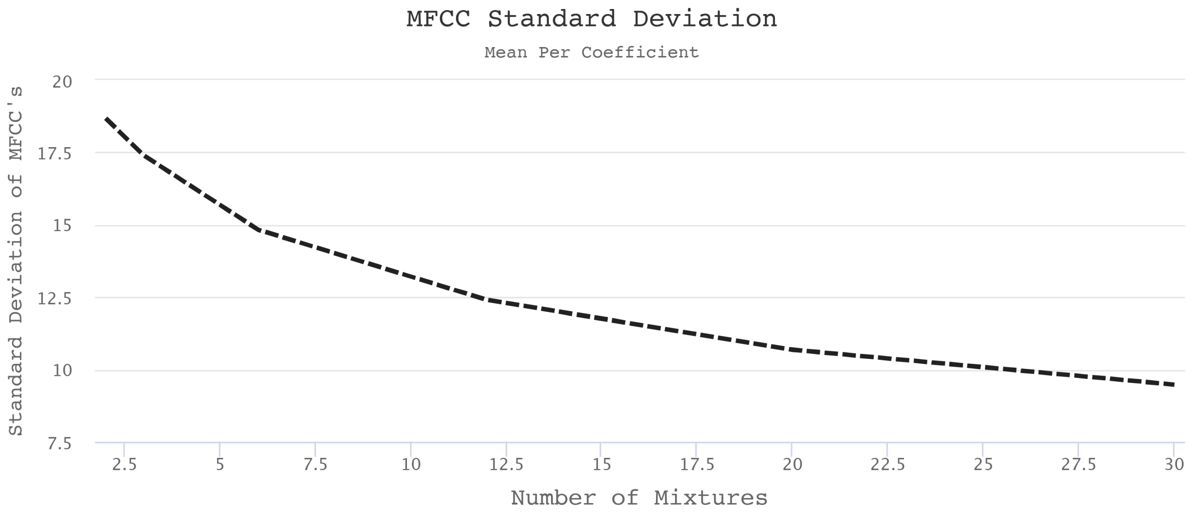


Figure 3.1: Standard Deviations (averaged across all 19 coefficients) of the measured MFCC coefficients of mixtures of audio files. As M increases, the variance in the MFCC coefficients goes down.

Librosa library [66] and compute the features on audio files sampled at 22050 Hz using the default window size (2048 samples) and hop size (512 samples) of the Librosa implementations. We then follow [51] in flattening both the FFT and MFCC features from 2-dimensional time-frequency representations into 1-dimensional feature vectors by taking a linear combination of the features at each frame, weighted by the RMS energy at the corresponding frame.

This method of averaging over time allows us to summarize the spectral characteristics of signals with different lengths using a single feature feature vector, while at the same time ignoring the quieter parts of the signal. In addition, we preserve the interpretability of the FFT and MFCC features through this process, which is particularly useful for inspecting and analyzing our model outputs. Of course, the downside of this preprocessing step is that we discard all time-domain information, so we are unable to predict anything about the envelope or movement of the sound. Depending on the source material and the downstream application, different preprocessing choices might be more appropriate than averaging over time; for example, unpitched percussive sounds require different modeling choices from pitched material. Since our data consists of mostly pitched notes from orchestral instruments, however, we follow the convention of the assisted orchestration literature by focusing on timbre independent of time.

Finally, before training or evaluating models, in order to best align our quantitative results to the expected perceptual results with regards to timbre, we normalize the FFT feature vectors such that the maximum value is 1. Although in the FFT case, relative magnitudes are known to be more correlated with perception of timbre than the raw amplitudes are, magnitudes of MFCC coefficients are important descriptors of timbre, so we do not normalize the MFCC's, instead predicting the real values.

Baseline

As a baseline, we compute the element-wise mean of the feature vectors over the entire training set. This vector is computed once for each value of the number of mixture components M. Models that perform better than this baseline can be said to be capturing some useful information about how the features sum together. One important factor to take into account when evaluating results it that as we increase M, mixing more and more notes together, the variance in the features of the mixtures decreases, making the predictive task appear easier. The MFCC features, because they are much lower dimensional than the FFT's, are especially effected by this change in variance; the higher dimensional FFT features exhibit the same trend but to a smaller extent, as they can capture a wider range of combinations of signals. For this reason, in Section 3.6, we report error metrics as a percentage relative to the error metric of this baseline at the corresponding value of M. Concretely, an error of 0.5 would mean that, averaged over the test set, the sum of squared errors of our predictions was equal to half of the sum of squared errors obtained by always predicting the mean of the data set.

Linear Combination

The first model we examine is the linear combination of features proposed in [63], which is currently used in state-of-the-art assisted orchestration systems [62]. This model implicitly assumes that for a feature f, the feature of the sum is approximately equal to the sum of the features:

$$w_1 f(S_1) + \ldots + w_M f(S_M) \approx f(w_1 S_1 + \ldots w_M S_M) \tag{3.1}$$

When features are linear or can be well approximated linearly, this method can be a strong baseline. Especially with high dimensional features like our 1024 FFT magnitudes, subtle details that might be difficult to summarize in an intermediate representation can be easily preserved with a linear model.

For this model, we combine source features in two ways, first by taking the element-wise mean of the M feature vectors as shown in Equation equation 3.2 and second by weighting the features by the corresponding RMS energies $a_1 \ldots a_M$ of the component signals as in Equation equation 3.3:

$$\bar{f} = \frac{1}{M} \sum_{f_i} \tag{3.2}$$

$$\tilde{f} = \frac{\sum f_i a_i}{\sum a_i} \tag{3.3}$$

MLP

Particularly for nonlinear features, it is reasonable to expect that nonlinear models have the potential to make better estimates. We train multilayer perceptron (MLP) neural networks to predict both FFT and MFCC features. For these models, we use a single hidden layer, and we minimize the mean squared error (MSE) between the predictions and the targets. For the FFT models, in order to constrain the network to output magnitudes between 0 and 1, we use a ReLU activation followed by a L_∞ normalization layer as the last stage in our network. Although we found empirically that sigmoid activations gave similar accuracies, these choices match better with the intuition of normalization performed in preprocessing. We train all our neural network models with Tensorflow [33] and Keras [67], Dropout [68], and the Adam optimizer [34].

Because we are interested in testing our methods on a variable number of audio mixtures M, we train separate MLP models for each value of M. As M increases, the input size and number of parameters in the network increases accordingly; with a feature of dimension D and a hidden layer of size H, the first layer of these networks has $D \times M \times H$ trainable parameters.

LSTM

As we increase the number of mixtures M, a recurrent network architecture is a natural choice to reduce the number of parameters needed. Intuitively, if a network can learn to estimate the sum of two signals, the same network should be able to process M signals in sequence over M steps by estimating the sum of one signal with the sum of all the signals processed so far.

Ordered Sets

Because the true sum of M signals is independent of the order in which they are combined, we experiment with two approaches inspired by the literature on sequence models for sets. First, even when no true ordering exists, previous results demonstrate that the ordering of inputs to factorized probabilistic models still affects the ability of models to learn [69]. In the case where two semantically valid orderings exist, empirical results from machine translation show that simple changes to the ordering, such as reversing the words in a sentence, can significantly affect model performance [11]. Based on these results, for this variant of the model, we sort the signals by their L_2 norm before passing them to LSTM model, such that the source signal with the highest energy is observed at the final timestep before outputting a final prediction.

Unordered Sets

Although previous work points to the benefits of ordering the signals in a consistent way, fixing an ordering prevents us from implementing a simple but potentially powerful form of data augmentation - randomly shuffling the order of mixtures during training. We empirically test the relative benefits of these two options, reporting results for both ordered and unordered inputs with the same LSTM archicture.

Residual Connections

Finally, we experiment with one more variation of our LSTM model, in which we add a residual connection [70] between the model inputs at each timestep and the outputs of the LSTM layer, which allows information to pass directly from the input to the final layer without having to be mediated by the nonlinear structure of the LSTM. Intuitively, to the degree to which features are linear, this connection should provide the model with the option to directly sum up the features as part of its computation.

3.6 Results

We train and evaluate all of the models across 6 different numbers of mixtures M ranging from 2 to 30, summarizing the results in Tables 3.1 and 3.2 and displaying the trends across values of M in Figures 3.2 and 3.3.

Predicting FFT Features

As demonstrated in Figure 3.2, the linear combination outperform the neural methods for values of M between 2 and 12, but the LSTM models make up ground and ultimately begin to overtake the linear combinations at $M = 20$ mixtures. All of the models in the FFT setting trend up in error towards the baseline as the number of mixtures increases; with $M = 30$, all models except for the residual LSTM cross the threshold of the baseline. These results indicate several findings:

- While the ordering in which the mixtures are passed to the LSTM model does not appear to make a significant difference here, the residual LSTM model outperforms the rest of the neural methods at all values of M, demonstrating increasingly large gains as the number of mixtures goes up. This suggests that the residual connection may be enabling the model to exploit the linearity of features when it is advantageous to do so, while maintaining flexibility to make better estimations once the signal from the linearity of the feature fades.

- In confirmation with previous findings [63], these results suggest that linear approximations of FFT features can be quite accurate. As the number of mixtures increases, however, these estimates worsen; by $M = 30$, the linear approximations are no better than random.

- Although estimating a high dimensional feature like the FFT is clearly a challenging task as many streams of audio are mixed together, these results show that neural models do possess the potential to estimate these features to some degree even in settings with many different sources.

Model	Mix 2	Mix 3	Mix 6	Mix 12	Mix 20	Mix 30
Baseline	1	1	1	1	1	1
Linear (Mean)	0.44	0.60	0.73	0.85	0.99	1.16
Linear (Energy-Weighted)	**0.15**	**0.25**	**0.41**	**0.62**	0.83	1.04
MLP	0.72	1.10	1.23	1.24	1.17	1.36
LSTM (Unordered)	0.54	0.69	0.78	0.81	0.89	1.34
LSTM (Ordered)	0.55	0.73	0.85	0.88	0.86	1.35
LSTM (Residual)	0.49	0.67	0.84	0.85	**0.81**	**0.91**

Table 3.1: Mean Squared Error for predicting FFT features across mixtures.

Predicting MFCC Features

Unlike in the case of the FFT features, all of the neural models outperform the linear combination for both small and large numbers of mixtures, and as shown in Figure 3.3, with more than 6 mixtures, linear combinations of MFCC features no longer contain a useful signal. We detail our findings from the MFCC experiments below:

- Because MFCC features are nonlinear, it is not surprising that nonlinear models are able to predict them better than the linear combination. Relative to the baseline, however, we can see that for mixtures of 2, 3, and even 6 different sources, a linear combination of MFFC's can still be reasonably accurate. This suggests that in some cases, MFFC features do behave approximately linearly when summed.
- In contrast to the FFT setting, the residual LSTM does not appear to offer any gains in comparison with the other LSTM models. Perhaps because of the much smaller dimension of the features, the Unordered LSTM model, which we train with data augmentation by randomizing the order in which mixtures of processed, performs best.
- As M continues to increase, the accuracies of the LSTM models flatten out rather than continuing to approach the baseline. This trend suggests that even when dozens of notes are mixed together, we may be still able to estimate certain features of these mixtures based only on the features of the source files.

Computation Time

While the exact computation time of FFT or MFFC features depends on the implementation, the length of the audio files, and the availability of parallel processing, estimating features with the networks we explore is, in practice, significantly faster than computing the real features. Though it is beyond the scope of this chapter to report results on a comprehensive list of hardware and software configurations, as a point of reference, Table 3.3 displays running times for parallel computation on our research server containing 20 CPU's and one Tesla K40 GPU.

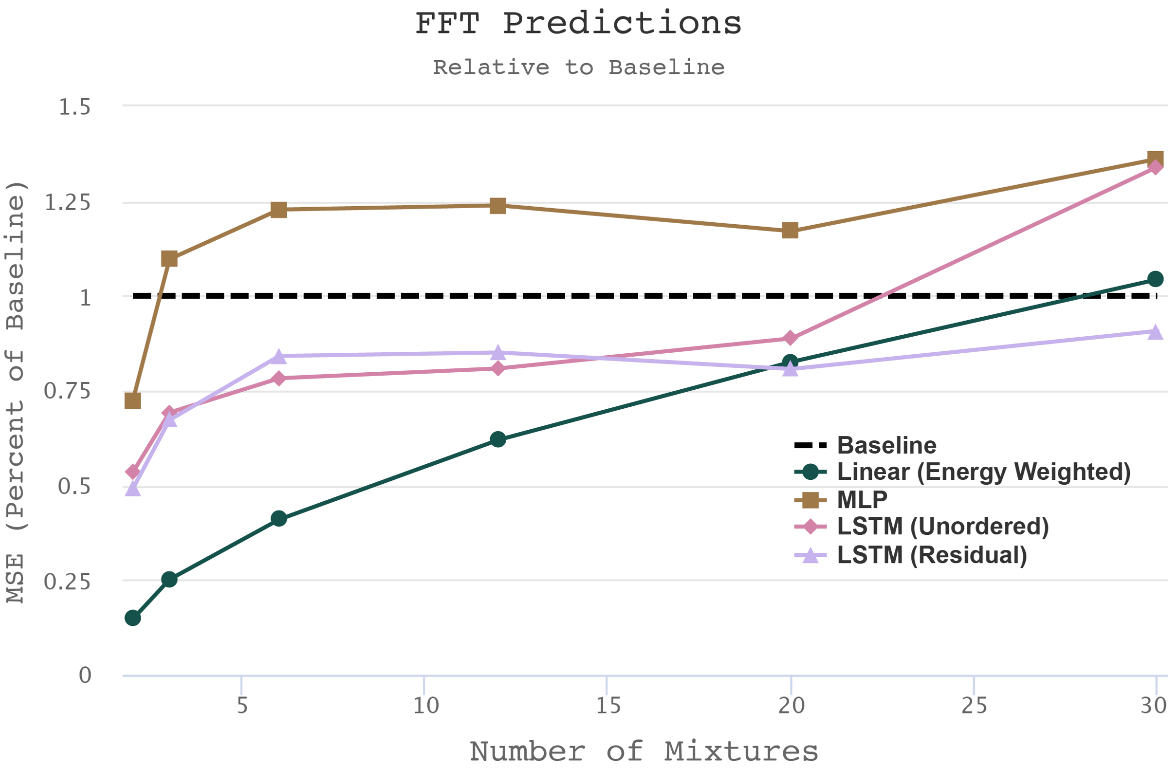


Figure 3.2: The linear models work well for predicting FFT features of smaller numbers of mixtures, but at around $M = 20$ mixtures, the best performing LSTM model overtakes the linear combination.

Model	Mix 2	Mix 3	Mix 6	Mix 12	Mix 20	Mix 30
Baseline	1	1	1	1	1	1
Linear (Mean)	0.43	0.58	0.81	1.03	1.32	1.54
Linear (Energy-Weighted)	0.36	0.59	0.94	1.30	1.69	2.02
MLP	0.42	0.55	0.71	0.79	0.88	0.93
LSTM (Unordered)	**0.30**	**0.46**	**0.57**	**0.63**	**0.71**	**0.70**
LSTM (Ordered)	**0.30**	**0.46**	0.61	0.66	0.73	0.73
LSTM (Residual)	0.32	0.47	0.64	0.71	0.77	0.77

Table 3.2: Mean Squared Error for predicting MFCC features across mixtures.

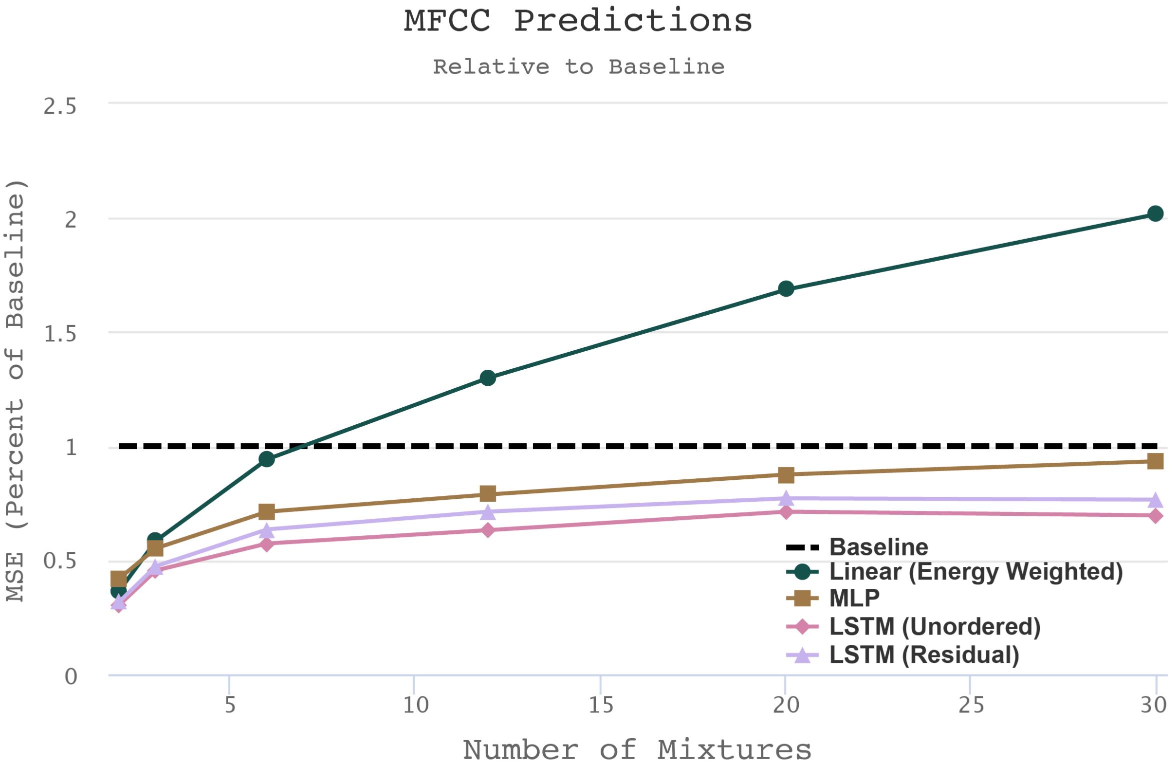


Figure 3.3: The neural models outperform the linear combinations significantly, widening the gap as M increases.

Feature	**Real** (CPU x 20)	**LSTM** (CPU x 20)	**LSTM** (GPU x 1)
FFT (Mix 2)	14.71	0.32	0.07
FFT (Mix 30)	14.71	4.75	1.10
MFCC (Mix 2)	73.50	0.03	0.01
MFCC (Mix 30)	73.50	0.34	0.15

Table 3.3: Time in seconds to compute or estimate energy-weighted FFT or MFCC features for the 2000 audio files in the test set using parallel processing. FFT (Mix 30) refers to the FFT feature of a mixture of 30 audio files, which requires 30 autoregressive LSTM steps. LSTM refers to the Residual LSTM model.

3.7 Conclusions

In this chapter, we experiment with neural models for predicting unobserved audio features based on precomputed features of source files in a mixture, examining the cases of FFT features, which should behave linearly when summed, as well as MFCC's, which are known to be nonlinear. We find that in the case of nonlinear features, LSTM models significantly outperform the methods currently in use for feature estimation, and further, that while the linear predictors perform well for small numbers of mixtures, as we mix more and more signals together, the neural models begin to outperform the linear methods as well.

Our results suggest that we may be able to improve current assisted orchestration systems [62] by replacing feature estimation components with LSTM-based nonlinear predictors. As with any real-world problem that involves perceptual similarity rather than comparisons in a feature space, however, more work is needed to understand how these models may interact with other components of systems they may be embedded in. Deep neural network models can and do adapt to any correlations present in the data, so understanding how these models are making there estimates may be important.

Beyond tasks like assisted orchestration in which we cannot always observe the features of an audio file because of computational limitations, we hope that future work may be able to take advantage of the methods for feature estimation explored here in order to make creative use of data like the Million Song Dataset, for which precomputed features are available but raw data cannot be distributed.

Acknowledgments

The research reported in this chapter was supported by the Hellman Family Faculty Fund and by resources provided by NVIDIA. We also thank the anonymous reviewers at ISMIR for their valuable feedback.

Chapter 4

What to Play and How to Play it: Guiding Generative Music Models with Demonstrations[1]

Abstract

This chapter proposes and evaluates an approach to incorporating multiple user-provided inputs, each demonstrating a complementary set of musical characteristics, to guide the output of a generative model for synthesizing short music performances or loops. We focus on user inputs that describe both "what to play" (via scores in MIDI format) and "how to play it" (via rhythmic inputs to specify expressive timing and dynamics). Through experiments, we demonstrate that our method can facilitate human-AI co-creation of drum loops with diverse and customizable outputs. In the process, we argue for the interaction paradigm of *mapping by demonstration* as a promising approach to working with deep learning models that are capable of generating complex and realistic musical parts.

4.1 Introduction

Communication between musicians can take time, effort, multiple attempts and clarifications, and often requires trial and error. In performance, composition, or production environments, contributors need to explain what they want from each other; any partnership or collaboration depends on the ability to clearly communicate ideas to the person whose job it is to execute those ideas musically (e.g. by playing an instrumental part, arranging a score, setting the level of a reverb effect, and so on).

When musicians and composers work with complex musical instruments and tools, communicating ideas to a machine can also require effort, exploration, and expertise

[1] The material is this chapter draws upon my previously published work in *What to Play and How to Play it: Guiding Generative Music Models with Multiple Demonstrations* at the 2021 New Interfaces for Musical Expression Conference [71] with co-author David Bamman.

(albeit expressed in a much different form), especially when the details of how an instrument works are opaque. Musical instruments and tools based on Artificial Intelligence (AI) and Machine Learning (ML), especially those built on powerful generative models capable of synthesizing human-like audio or MIDI, can be particularly difficult for users to navigate in predictable ways. Still, realistic and expressive outputs from this kind of model have inspired a growing interest among music creators to explore incorporating generative ML models into their creative practices [1, 72, 73].

Recent research highlights that while music creators can often count on ML music models to provide them with surprising or unexpected ideas, they tend to have a hard time *controlling* them, finding it difficult to achieve specific results when desired [73, 74, 75]. In response, a number of recent studies seek to make generative models easier for users to control by making them *conditional* - by training models with different types of input variables as probabilistic conditioning.

In practice, inputs to conditional generative models can take many forms, for example categorical variables like genre or the identity of a specific artist [1], initial themes for continuation [76, 18, 77], pitch contours [78, 79, 80], chord symbols [10, 81, 82], accented rhythms [6], or features summarizing the characteristics of individual notes [83, 84]. Once a model has been trained, these variables can be exposed in different ways within user interfaces to provide different affordances. Before reaching this stage, however, the choice of conditioning variables (along with the choice of training data) outlines an initial set of limitations that define how a model might be used.

If our intended use for a generative model is to provide inspiration, to help us break out of existing patterns or habits, or to challenge ourselves by including a "musical other" into our composition practice [73], then many different ways of conditioning a model may serve us well; indeed, other approaches that do not involve ML may also work just as well. As soon as we begin to make our goals more specific, however, designing and implementing conditional models becomes harder [74] and requires solving interconnected technical and interaction challenges at the same time.

In this chapter, taking inspiration from the ways in which musicians communicate with one another - in particular, by demonstrating an idea with multiple views drawn from different modalities - we contribute and experiment with a framework for designing and training conditional generative models with multiple complementary user inputs.

To anchor this notion of communication through multiple demonstrations with a specific recorded example, consider the diverse array of communication styles displayed by music producer Oak Felder in the process of collaborating with a drummer [85]. Within the span of no more than a few minutes, Felder: **(1)** offers high-level stylistic suggestions ("I'm wondering if it should be a little more complex."), **(2)** provides specific instructions about one instrumental part ("No hi-hat."), **(3)** demonstrates a drum pattern through sound with a vocal imitation, **(4)** indicates a drum fill by briefly playing air drums, and **(5)** nods his head to the side in time with the music to show where accented beats should go. Some of this guidance is given through examples (e.g. vocal and motion-based gestures), and other instructions, though expressed verbally by Felder, could presumably also be demonstrated to a machine by example (e.g. a blank hi-hat track indicates “No hi-hat.”) Over the course of this interaction, Felder conveys some of the more concrete details only once (e.g. “No hi-hat”), while reinforcing more abstract concepts by demonstrating them in more than one way (e.g. gesturing a drum pattern in the air while vocalizing a version of it at the same time). In the end, based on all these different cues, the drummer picks up on the intentions of the producer, and they successfully record the part together.

We do not bring up this example in order to argue that we should interact with computer models just like we do with humans, using natural language interfaces and so on; rather, we find it instructive to highlight the range of examples that a producer instinctively draws on here in order to convey their intention to the drummer. By breaking down an idea, which at firsts only exists in Felder’s imagination, into complementary (even if sometimes overlapping) components, some of which can be expressed well in one way and some better in another, the producer can convey information to the drummer more effectively.

Drawing inspiration from this kind of multifaceted communication between producer and musician, which happens not instantaneously but over the course of the time it takes to design or perform the relevant demonstrations, we experiment in this chapter with building generative models that accept two or more user-provided conditioning inputs given *by example*, with each input designed so as to be possible for a user to create. ML models offer promise as useful tools particularly when a user has an idea in mind that is difficult to create from scratch (for example because the user is not sitting in front of a drum kit or doesn’t know how to play drums [6], but which can be still be specified by example in some simpler form.

To ground our experiments in a context that we hope can be of practical use to music creators, we focus on models for generating two-measure drum loops. This particular task of creating drum and percussion parts is of broad interest to creators in many styles of music, and models for generating drums have already been implemented in publicly available toolkits for music producers [75, 86, 87], making it easier to implement the methods we explore within interfaces similar to those in the toolkits above. Using drum recordings from the Groove Midi Dataset [6], we explore Variational AutoEncoder (VAE) models [88] for generating drum beats based on two or more user inputs, with every input defined in MIDI format and able to be specified by example either through gestures recorded by a MIDI keyboard or microphone, or through grid interfaces like those found in drum machines. In working on creating drum loops, we pay particular attention not just to the pattern of which drums are to be played, but also to how they are played, modeling precise microtiming and dynamics information, which is known to be difficult for users to create by hand *without* performing any gestures to demonstrate.

This chapter’s primary contributions are as follows:

- We design and implement a factorized Variational AutoEncoder model for generating drum performances conditioned on multiple inputs that cover aspects of both a musical score and how that score should be played. We experiment with a model that accepts two inputs and another that takes up to eleven, more fine-grained, inputs. We demonstrate that these models allow us to generate drum loops with more diverse and more precisely specified outputs than existing methods.

- We show that by factorizing score and performance characteristics into separate latent variables, we can overcome difficulties with sampling encountered in previous work in order to maintain diverse outputs while still leveraging efficient data representations that use continuous rather than discrete values to model microtiming and dynamics in music.

- We tie together recent research in conditional generative models for music with the interaction framework of *mapping by demonstration* and offer a technical approach based on models that can accept multiple demonstrations from users, which we hope will take steps toward enabling future user-centered research on human-AI co-creation with music generation models.

Code and pre-trained models developed in this chapter can be found at: `https://github.com/jrgillick/groove`

4.2 Related Work

This chapter builds on previous research on drum loop generation from Chapter 2, which serves as a starting point for the applications and the machine learning methods explored here. Previously, we proposed two models for conditional generation of drum loops using a Recurrent Variational AutoEncoder (a GrooVAE). One model explores the task of *Humanization* - automatically generating dynamics and timing variations giving a quantized Midi input, and the other proposes an application called *Drumify*, in which a model generates drum loops based on an input rhythm with expressive timing (which could be tapped out on a surface or implied by the onsets of another instrument), but with no specified instrumentation or score. In each case, these models are able to synthesize realistic drums that listeners have difficulty distinguishing from real loops in the data set.

Both of these interactions, however, are limited in that they only afford the user one input at a time in order to specify what they want. This means that in practice, if a user has a specific beat in mind, the *Humanize* model does not offer control over *how* the model will add expressive dynamics and timing to that beat; as a result, for any given input score, the output is almost always the same. Similarly, the *Drumify* model does not provide any control over which drums are played; for example, it is left up the model to choose whether to use the ride cymbal or the hi-hat. In our experiments here, we attempt to address these limitations with regard to both diversity and control.

We also draw more broadly from a number of other studies on conditional models for music generation. Recent work on music generation based on some kind of user input includes models that provide counterpoint to an improvised melody [89], map eight buttons on a game controller to the 88 keys on a piano [90], or synthesize the audio for one instrument based on fine-grained pitch contours and dynamics from another signal [80]; we build on these by exploring multiple complementary gestural inputs at the same time. On the modeling side, we also build on work using factorized representation learning to control generation of monophonic [91] or polyphonic [78, 81] music scores. We explore a different kind of factorization here, however, by separating out scores from performance characteristics, as well as a different model architecture.

Finally, we draw inspiration from gesture mapping [92, 93, 94] in designing the conditioning inputs used in generative models around the concept of a *gesture* (which has been defined in a number of different ways but can be broadly categorized as some kind of sensed input performed or specified by a user). Much research within the NIME and Computer Music communities focuses on interaction paradigms centered around mapping various kinds of user inputs (which often take the form of performable gestures) onto output parameters for controlling sound [93, 95]. By providing demonstrations of gestures, users can train their own mapping models by example using machine learning [94]. Most approaches to gesture mapping attempt to modify a relatively small number of output parameters (e.g. a handful of knobs on a synthesizer) [96], as opposed to the many thousands or millions of parameters in large neural network models; as a result, gesture mapping often provides more precise control than has typically been possible with large music generation models.

In addition to the large number of parameters to learn, another barrier that has inhibited music generation models from being put to use in the same way as gesture mapping is the size of datasets and expense of computational resources needed to train them, which prevents users from choosing and manipulating their own training data. A number of recent studies have explored ways to either make models smaller and faster to train [87] or to enable customization of pre-trained models to meet user needs [97]. We see this line of work as complementary to the model conditioning work that we explore here; depending on the context, interactions may be better facilitated by more precise conditioning controls, easier management of training data, or a combination of both.

4.3 Proposed Models and Implementation

Modeling Two Inputs: Score and Groove

Starting from the hypothesis that multiple different forms of user input can lead to more controllable and diverse generated music, we operationalize the idea of model inputs as gestures by implementing a factorized neural network model architecture called an Auxiliary Guided Variational AutoEncoder [98]. We first implement a model that accepts two inputs - one for quantized drum scores (specifying what to play) and one for tapped rhythmic inputs (specifying how to play it), with each of these inputs implemented exactly as in the previously published *Humanize* and *Drumify* models [6]. An important point to make here is that these inputs are not directly provided in the data set; at training time, as with other AutoEncoder models, we are restricted to using

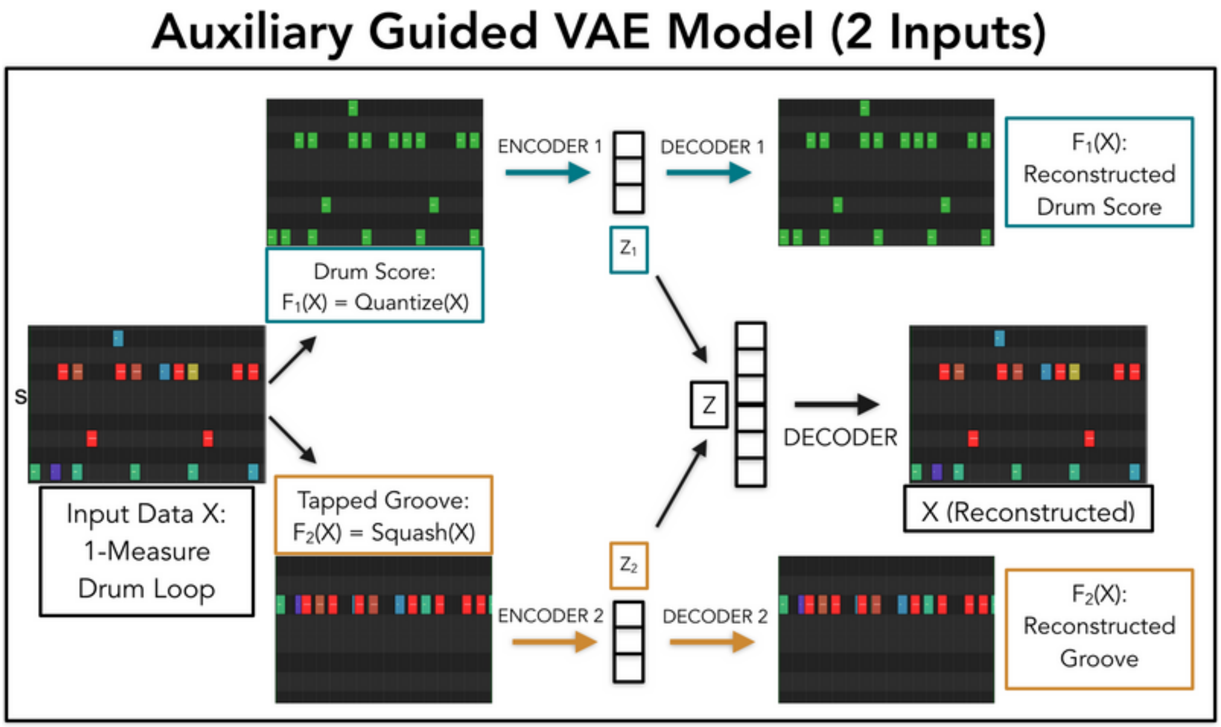

Auxiliary Guided VAE Model (2 Inputs)
ENCODER 1
DECODER 1
F₁(X): Reconstructed Drum Score
Drum Score: F₁(X) = Quantize(X)
Z₁
S
Z
DECODER
Input Data X: 1-Measure Drum Loop
Tapped Groove: F₂(X) = Squash(X)
X (Reconstructed)
Z₂
ENCODER 2
DECODER 2
F₂(X): Reconstructed Groove

Figure 4.1: Auxiliary Guided Variational AutoEncoder model trained to take two user inputs (a quantized drum score and a tapped rhythm expressing the groove of the loop). Features of the drum sequence, which are designed to be similar to inputs that could be demonstrated by a user through an example, are extracted via functions $F_1(X)$ (here, a quantization function that removes microtiming and velocity) and $F_2(X)$ (here a "squashing" function that preserves microtiming and velocity but discards the score).

inputs that can be computed with some function F applied to an input data point X. Through the design of a function $F(X)$, we specify a mapping from drum loops (high dimensional realistic data points) to simplified descriptors of those loops (which are easier for users to create with a gesture); we then train models to learn inverse mappings from gestures to data. For this model, we define two functions during training that take the place of user inputs at inference time: $F_1(X)$ is a quantization function that removes all microtiming and velocity information from a drum loop (keeping only drum score), and $F_2(X)$ is a "squashing" function that has the opposite effect, keeping performance characteristics in the form of microtiming and velocity, but discarding the drum score. Figure 4.1 visualizes the architecture of this neural network model.

This architecture differs from a standard VAE in two ways. First, while a typical VAE, which we treat as a baseline, has a single latent variable Z, the *Score* and *Groove* inputs to this model are each encoded (in this case with bidirectional LSTM encoders) into separate latent variables Z_1 and Z_2, which are both independently trained to match standard normal distributions; following Roberts et al. [26], we train using the free bits method (hyper-parameters to balance the two loss terms in a VAE) with a tolerance of 48 bits. Z_1 and Z_2 are subsequently concatenated and passed to a decoder (also an

LSTM), whose goal is to reconstruct the original drum sequence from the training data. This separation between Z_1 and Z_2 (sometimes called factorizing or disentangling) aims to explicitly capture some of the variation among each of these two aspects of the data (*Score* and *Groove*) with specific variables. One of our goals of factorizing in this way is to attempt to overcome problems with diversity reported in previous work, in which when generating performance characteristics for an input score, a given loop was always *Humanized* in the same way [6]; with this model, by sampling different values for Z_2 or inputting different *Grooves*, we can try to synthesize different performances for the same score. This factorization also affords a user two complementary ways of specifying a desired drum loop (by independently providing a *Score* and a *Groove*).

The second distinguishing feature of this architecture is the inclusion of Auxiliary Decoders, similar in form to those proposed by Lucas and Verbeek for image generation [98]. As shown in Figure 4.1, in addition to the decoder trained to reconstruct the original drum loop, separate decoders ($Decoder_1$ and $Decoder_2$) are trained at the same time to reconstruct the input *Score* and the input *Groove*. This variant of an AutoEncoder, which appears not to have been employed before for modeling music, offers promise for two reasons: first, it explicitly reinforces the incentive for the latent variables Z_1 and Z_2 to capture the relevant information, and second, it offers a mechanism for users to inspect the model's interpretation of each input gesture: along with a generated drum loop, a user can also listen to or visualize the model's reconstructions of the *Score* and the *Groove* corresponding to that loop. Examining these auxiliary reconstructions allows the user (or model developer) one option for investigating the strengths and weaknesses in the model, which may be helpful in learning how to work with it. For example, if the auxiliary reconstruction of a user-provided *Groove* is inaccurate, this suggests that the model is unable to recognize the given gesture; this feedback can direct the user to try again by performing the gesture slightly differently in order to better work within the model's limitations.

Breaking it Down Further: Modeling More Inputs

In addition to the VAE with two inputs, in the spirit of our motivating example where a producer explains a drum beat to a drummer in several different ways, we further experiment with factorizing our model into more components, with the hope of capturing more options for diverse outputs and controllable interaction. Here, we divide the latent variable Z into 11 components $Z_1 \ldots Z_{11}$. This time, we separate the 9 different drum instruments from the score into 9 different latent variables (visualized at the top of Figure 4.2), such that a user can specify as few or as many of these as they choose to, with the option to sample the others. For example, a user can specify a pattern for the kick and snare drums, provide an empty pattern on the crash cymbal channel indicating not to play any crash cymbals, and through sampling the other latent variables, leave the choice of whether to add hi-hats or ride cymbal for the model to decide. At the same time, in addition to the *Groove* input defined in the first model, we add a second performance-style input that captures musical *Accents*, indications of where notes are emphasized by being played louder. Here, we define accents as binary vectors with

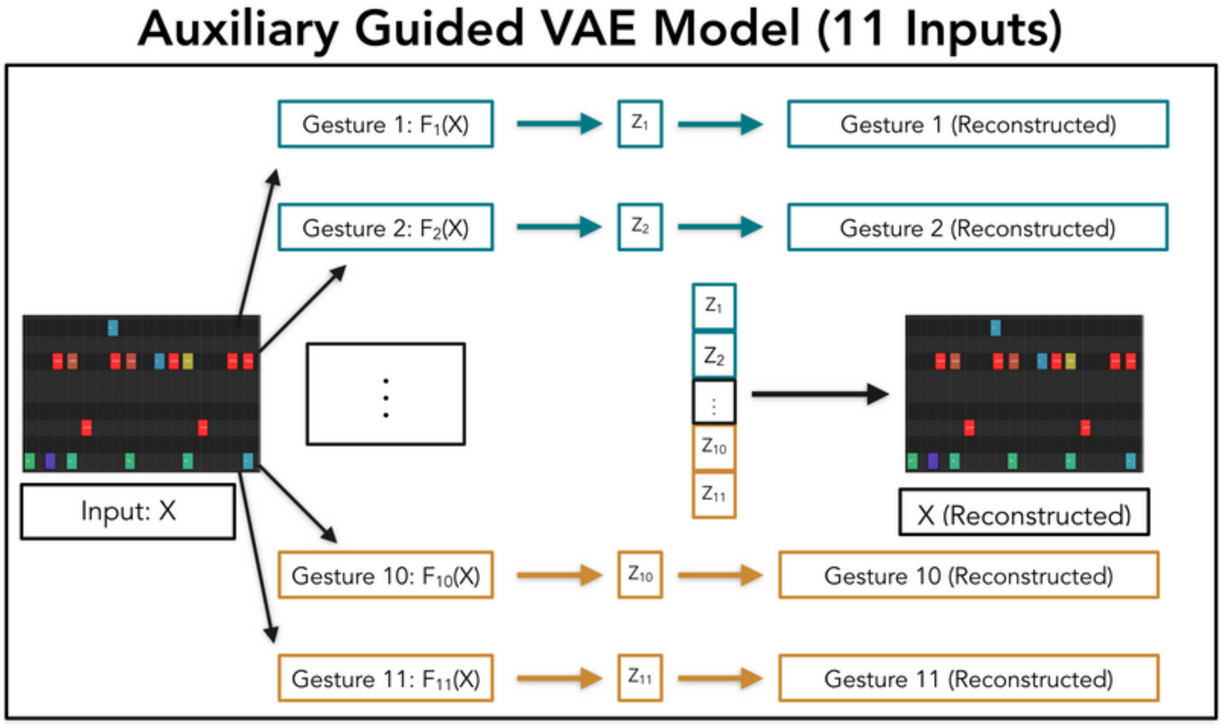

Auxiliary Guided VAE Model (11 Inputs)
Gesture 1: F₁(X)
Z₁
Gesture 1 (Reconstructed)
Gesture 2: F₂(X)
Z₂
Gesture 2 (Reconstructed)
Z₁
Z₂
⋮
Z₁₀
Z₁₁
Input: X
X (Reconstructed)
Gesture 10: F₁₀(X)
Z₁₀
Gesture 10 (Reconstructed)
Gesture 11: F₁₁(X)
Z₁₁
Gesture 11 (Reconstructed)

Figure 4.2: Auxiliary Guided VAE Model with 11 Inputs. This version breaks drum loops further into 11 different latent variables: 9 based on the score (1 for each instrument in the kit) and 2 based on performance features (one specifying microtiming through a tapped rhythm and one specifying accented beats).

one input corresponding to each 16th note timestep (we use a 16th note resolution in time for these models, although other resolutions offer different advantages and disadvantages [87]; we consider a metrical position in the dataset to be accented if it contains a note (on any drum) whose Midi velocity is more than one standard deviation above the mean velocity for that drum, calculated over the entire sequence.

In describing our models, we adopt the terminology of *gesture* to refer to each of the inputs, though some inputs could be either performed by a user in the typical sense of a gesture, or created in another way, for example by composing them in a Midi editor. In this second model, because each gesture is expected to be packed with less information, we simplify the encoder and decoder architectures in the interest of reducing model size and training time, using small feed-forward MLP neural networks instead of LSTM. We experimented with simplifying the main decoder as well, but we found that in order to generate realistic outputs comparable to those in previous work, it was important to use a more powerful architecture than an MLP, so we use an LSTM here as well.

4.4 Experiments

To evaluate our model designs, we examine metrics computed on the test partition of the Groove Midi Dataset, measuring two main aspects of our proposed methods. First, we look at the diversity of generated drum loops using our models, comparing against the aforementioned GrooVAE model [6] in the context of the task of *Humanizing* quantized drum scores (by generating MIDI velocities and microtimings), and second, we examine the potential for controllability afforded by these models. While controllability will ultimately depend on the context of how, and with which users, a model is situated in an interactive setting, as a starting point, we use the idea of *fidelity* as proxy: given a particular input gesture, we examine the degree to which the model outputs exhibit the characteristics demonstrated by that input.

We have not yet deployed these models in an interactive interface to study their usability in practice, but this choice of metrics is informed by our previous findings in which we deployed the *Humanization* and *Drumify* models (treated here as baselines) as plugins in Ableton Live [6, 75] and tested them with users. We believe that improving on these quantitative metrics is an important next step in our ongoing iterative process of designing tools for musical human-AI co-creation.

Measuring Diversity in Generated Performance Characteristics

To measure diversity, we explore the task of *Humanization*. In this task, a model's job is to take a quantized drum loop as input (a *Score*), and then synthesize performance characteristics (microtiming and velocity) for that input. One of the motivating factors for exploring the work in this chapter was the shortcoming of our baseline model proposed in Chapter 2, which, although able to create realistic outputs, always generated the same stylistic outcomes. For this metric, we look at the standard deviations of timing offsets generated by each model. Following the baseline implementation, we calculate timing offsets as continuous numbers between -1 and 1, which represent how far each drum onset falls between the current timestep and an adjacent one. Drum hits played late, or behind the beat, are represented by positive numbers here, and drum hits played early, or ahead of the beat, are given negative numbers.

Using two-measure windows taken from every drum performance in the test set (a total of 2192 sequences), we *humanize* each drum sequence five times with each model, and then among each set of five generated loops, we compute the mean element-wise standard deviations of the timing offsets, such that notes in the same position (e.g. a snare on beat 3) are compared with each other. This yields a single measurement for each test sequence, which we finally average across the entire test set. A higher standard deviation here indicates more diverse outputs.

In this experiment, we compare three conditions: **(1)** the baseline Variational AutoEncoder model that includes neither factorized latent variables nor Auxiliary Decoders, **(2)** our factorized model without Auxiliary Decoders, and **(3)**, the full model shown in Figure 4.1. In the baseline model, only the *Score* input is provided; for our new models, we implement the *Humanization* task by taking a single score as input, while, across each of the five runs, we sample a random vector for Z_2 to pass to the decoder.

Measuring Fidelity to a Gesture

In a second experiment, as a proxy for measuring the controllability of interactions with our models, we look at how well the generated outputs match the characteristics of a given gesture in the new model. Here, we fix an input *Groove* with a pre-specified pattern of timing offset values (e.g. 0.5 for every off-beat 16th note and 0 for every on-beat 16th note to indicate a 16th note swing), before applying the same *Groove* to every *Score* in the test set using the 2-input Auxiliary Guided VAE model shown in Figure 4.1. After applying the same 3 *Grooves* as conditioning inputs paired with the *Score* extracted from every sequence in the test, we plot the resulting distributions for each groove and measure the means and standard deviations of the generated timing offsets on the off-beats. For the three different input grooves, we use a different fixed offset value (-0.05, -0.2, and -0.4, respectively) for every alternate 16th note position. This corresponds roughly to choosing a particular *Swing* value (as is common in drum machines) as a conditioning input. Unlike drum machines, however, in which timing offsets are applied uniformly through a templated approach, we might not expect the synthesized outputs from our machine learning models to conform exactly to this value; the goal here is again to guide the model towards a particular groove rather than to control it exactly.

4.5 Results and Discussion

Through our quantitative evaluations, we find that, in general, the methods explored in chapter work appear promising for both diversity and controllability in generated drum loops. As Table 1 shows, our measurement of diversity confirms the finding reported previously that the baseline model usually performs *Humanization* in the same way each time. The Standard Deviation metric of 0.061 (measured as a proportion of the distance between successive metrical positions as 16th note resolution) for the baseline in Table 4.1 is quite small; for context, even changing the timing of a drum pattern by two standard deviations here would not be enough, for example, to change a beat from a straight feel to a heavy swing feel. The factorized VAE models, however, show a different trend, with much higher Standard Deviations among the timing offsets; the version using Auxiliary Decoders shows the most diversity here with a Standard Deviation of 0.22. Furthermore, alternative methods for adding diversity during sampling do not help the baseline here: increasing the value of the temperature parameter in the decoder does not change the metrics in Table 4.1, and adding noise to the latent vector Z before decoding has the undesirable side effect of causing the model not to follow the given input *Score*.

Our subjective experience in listening to these *Humanizations* accords with the metric here as well; we find that unlike with the baseline, these models generate perceptu-

Model	Standard Deviation of Timing Offsets
Baseline VAE [6]	0.061 ±0.001
Factorized VAE	0.200 ±0.002
Factorized VAE + Auxiliary Decoders	**0.222** ±0.002

Table 4.1: Measuring Diversity in Generating Timing Offsets

Target	Generated (Mean)	Generated (Std. Dev.)	Difference in Means
-0.05	-0.091	0.161	0.042
-0.2	-0.214	0.163	0.014
-0.4	-0.366	0.175	0.034

Table 4.2: Measuring Fidelity to a Gesture (Swing Amount)

ally different results. Depending on the *Groove* conditioning, sometimes the same beat is played with a swung or triplet feel, and other times it is played straight. In addition, drums and metrical positions are accented different across different runs.

In our second experiment, a case study in examining the fidelity of our Auxiliary Guided VAE model to a gesture (the gesture in question is a *Groove* representing a particular amount of swing), we find that when applied broadly to a large number of input *Scores*, the average swing values (as measured by timing offsets on off-beats) come quite close to the target values. Different swing values lead to slightly different trends here: guiding the model toward more heavily swung beats tends to give slightly larger variation in the generated outputs than when specifying beats with less swing, and in general, offset values tend to regress slightly to the mean of the entire dataset. Table 4.2 summarizes these results, and Figure 4.3 visualizes the distributions from this experiment.

In addition to the metrics reported above, which focus on the 2-input model factorizing *Score* and *Groove*, we also explored the larger 11-input model more informally by

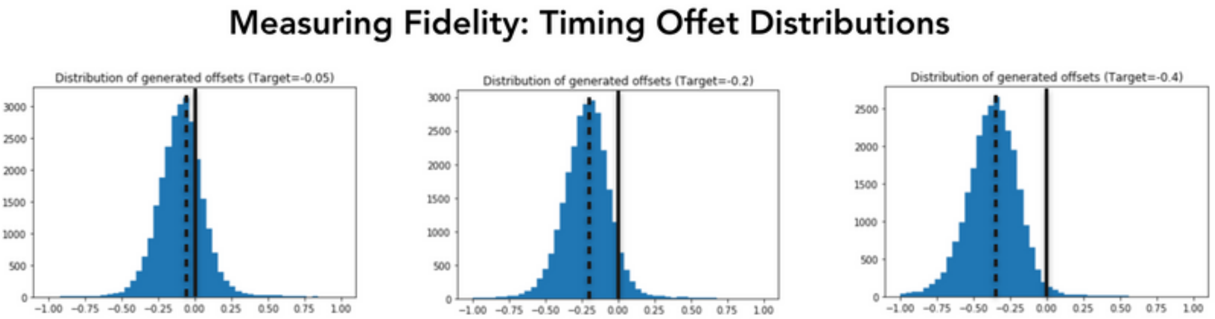


Figure 4.3: Distribution of timing offsets for 3 different target *Grooves*. From left to right, the target values specified by the three conditioning inputs are -0.05, -0.2, and -0.4.

listening to a number of outputs with different conditioning setups. For example, in one experiment, we fixed all of the gestural inputs except for the features specifying the intended patterns for hi-hats and ride cymbals. We then applied several different hi-hat or ride input patterns given the same fixed set of other conditioning inputs. We found that the results were usually quite realistic, though in some cases slightly less so than with the baseline or the simpler 2-way model. The possibilities for diversity and control, however, appear richer: the model did follow the input specification, reliably switching between hi-hat and ride cymbal, while still following the same groove in each alternate condition. The model also seemed to make reasonable choices in this case when forced to choose between mismatched conditioning inputs (e.g. specifying an *Accent* or emphasizing a *Groove* in a metrical position where the corresponding score is blank). As we might expect, however, not all combinations of input gestures are able to lead to realistic results; in particular, when we specified less common patterns through the input gestures, model outputs were either less realistic or less faithful to the specified gesture.

4.6 Conclusions

Designing and developing technology to guide complicated generative music models towards user-specified musical goals is a challenging problem that has recently seen increased interest among ML and MIR research communities. As researchers from these communities have increasingly turned away from working solely on the technical aspects of machine learning and toward studying how to making generative models easier for users to control, research questions have begun to overlap with existing work on mapping from the NIME community: increased control, broader interaction possibilities, and new methods for human-AI co-creation motivate much recent work on conditional models for music generation [10] [15][23]. This convergence (which has also been raised by others [3]) motivates our current work, in which we aim to continue to move music generation research toward directions where it may be able to meet the creative needs of music creators.

In this chapter, we build on this strand of technical research, exploring a new combination of conditioning inputs and implementing them in a model for generating drum loops. At the same time, drawing inspiration from work in on gesture mapping, we reinterpret the technical formulation of conditional generative models into a simple interaction paradigm based on guiding ML models with demonstrations, and show through experiments as well as informal subjective evaluations that our approach can enable diverse and controllable interactions with music generation models. We hope that this approach will provide useful grounding for future technical and user-centered research on musical interactions between people and AI.

Chapter 5

Data Representations and their Impact on Music Creation[1]

Abstract

This chapter presents a new data representation for music modeling and generation called a Flexible Grid. The representation aims to balance flexibility with structure in order to encode all the musical events (notes or rhythmic onsets) in a dataset without quantizing or discarding any temporal information. In experiments with a dataset of MIDI drum performances, we find that when implemented in a Variational AutoEncoder (VAE) model, Flexible Grid representations can enable detailed generation of music performance data that includes multiple different gestures and articulations.

5.1 Introduction

One of the central affordances of music production and editing tools is the ability to place musical elements at precise positions along a timeline; many genres of music have emerged out of communities of artists working within the constraints of perfectly consistent rhythms and tempos. How and when to diverge from that grid is an important factor for creators to consider; many of the instrumental gestures that our ears are attuned to, like drumrolls, guitar strums, and trills, are composed of groups of rhythmic onsets that live in the spaces between the grid lines. Some music producers, like UK garage artist Burial, prefer to ignore the grid altogether [100], while others rely on setting global parameters like ”swing” [101]. Both of these approaches have their drawbacks: working completely without a grid is too time consuming for most to consider, and adjusting timing with global parameters offers only broad strokes rather than precise control.

[1] The material is this chapter draws upon my previously published work in *Drumroll Please: Modeling Multi-Scale Rhythmic Gestures with Flexible Grids* in the Special Issue on AI and Musical Creativity of the Transactions of the International Society for Music Information Retrieval Journal [99] with co-authors Joshua Yang, Carmine-Emanuele Cella, and David Bamman.

Given these limitations, one of the more intriguing directions offered by AI for expanding the rhythmic possibilities in music production is its ability to assist users in intelligently and subtly keeping their music ”off-the-grid” by modeling the rhythmic nuances of existing music. In the context of drums and percussion, several systems based on modeling instrumental performances have already been designed and made available within mainstream music production tools like Ableton Live [102, 86, 87].

Despite a long history of research in expressive performance analysis and generation (see [103] for a detailed review), generating expressive musical parts on the scale of even one or two measures in length remains a challenging problem.

Most research on expressive performance generation has been situated in the context of Western art music [103] and often relies on note-level alignments with scores or other structural elements of notated music like dynamics markings [104]. Some recent approaches based on deep learning have instead attempted to jointly model both composition and performance using MIDI data sourced from instruments outfitted with sensors (like a Disklavier or electronic drum kit), or from audio recordings automatically transcribed to MIDI [19]. Still, designing and engineering models that work well enough to generate compelling outputs requires large instrument-specific training datasets [19], compromising on temporal precision through varying levels of quantization [6], or both. To enable creators to explore the potential uses of expressive performance models in practice, we would like to be able to train music generation models with as little data as possible [87], while at the same time preserving the nuances of expressive music that can only be captured with precise temporal resolution.

In this chapter, by taking a close look at the representations used to encode drum performance data, we take steps to address some of the challenges that arise when modeling *off-the-grid* data with neural networks. We analyze the tradeoffs imposed by different representations, propose an alternative approach called Flexible Grids, and conduct experiments to investigate the relative advantages of each data representation.

Although the primary focus of this chapter deals with methodology — how to represent musical data when working with machine learning models — our work is motivated by the range of real-world applications that depend on these underlying mathematical representations of music. For this reason, in choosing our technical direction, we prioritize applicability toward directions that would otherwise be difficult for creators to explore (off-the-grid music), real-world constraints on data size and computational efficiency that are necessary for making AI broadly accessible [105, 87], and considerations of *interpretability* and *controllability* that matter to music creators when co-creating with AI [106]. Concretely, our contributions include the following:

- We analyze and compare existing data representations that have been used recently for music generation in the MIDI domain, highlighting opportunities for improvement.
- We present Flexible Grids (visualized in Figure 5.3 and described in Section 5.3) as an alternative data representation, along with motivations and implementation details.

- Using the Groove MIDI Dataset [6], a collection of drumset recordings containing expressive timing and dynamics, we experiment with training Variational AutoEncoder models using Flexible Grids as well as several other data representations, comparing results through quantitative metrics and a listening survey carried out with drummers. We also compare the same set of representations in classifying the anonymized identities of drummers in the dataset.

Code and trained models are available at: https://github.com/jrgillick/groove.

5.2 Data Representations for Musical Events

Recent work on music generation in the MIDI domain typically takes one of two broadly defined approaches to representing musical data. These two categories, which we will refer to throughout this chapter as Fixed-Grid and Event-Based representations, differ primarily in terms of how they handle musical time and tempo. While not all existing approaches fit neatly into one bucket or the other, this distinction is convenient for summarizing the main factors to consider when choosing a musical representation; [107] draw a similar distinction while connecting Fixed-Grid and Event-Based representations respectively with similar structures developed in Computer Vision and Natural Language Processing.

Fixed-Grid representations

Fixed-Grid representations break down music into equal chunks of time, typically associating each timestep with a musical duration such as an 8th note or a 16th note. As a consequence, musical constructs like tempo and beat subdivision can be built into Fixed-Grids, with time usually defined relative to a local or global tempo. This structure accords with theories of how humans perceive rhythm in that when multiple rhythmic onsets take place within a short time frame (a ”beat bin”), humans tend to group them together, hearing them as forming a single beat [108]. Tempo-relative representations are also advantageous for machine learning because they implicitly keep track of time, while at the same time outlining a shared structure for jointly modeling music recorded at different tempos.

Besides capturing some useful temporal musical structure, the other defining characteristic of Fixed-Grid representations is that they have a consistent *size*; this means that any sequence lasting one measure will always have the same number of timesteps (e.g. 16 or 32) and the same number of features per timestep, regardless of the density of musical events actually present in that sequence. Fixing the size of sequences is desirable for two main reasons: first, it enforces fewer constraints on the machine learning models and architectures we can choose from (feed forward and convolutional neural networks are a workable choice here), and second, maintaining an alignment with musical time allows us to design more predictable interactions: for example, we can

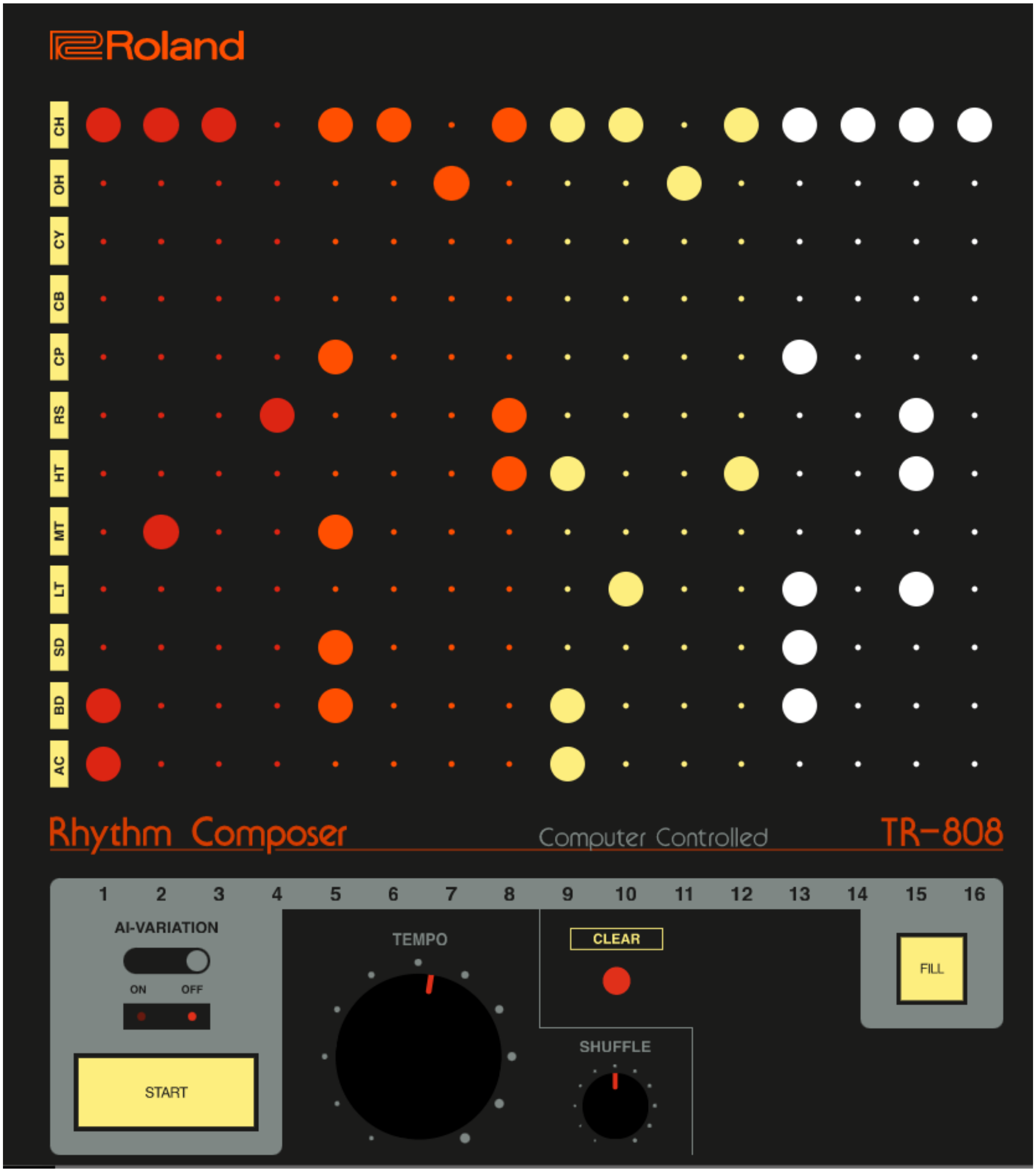


Figure 5.1: Fixed-Grid representation of a 1-measure pattern for 12 drums in a web interface designed by Yuri Suzuki (808303.studio) and inspired by Roland's TR-808 Rhythm Composer.

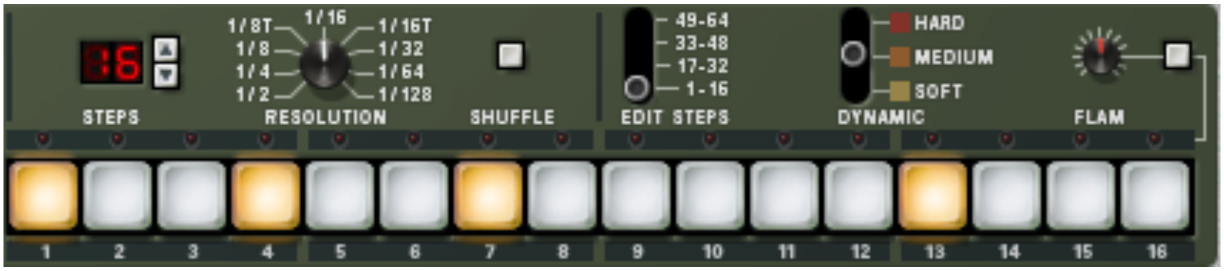


Figure 5.2: Fixed-Grid representation of a 1-measure pattern for a single drum in the interface for Propellerhead's ReDrum drum machine.

select, visualize, or manipulate musical parts that last for a specific number of beats or measures.

Formally, a Fixed-Grid representation consists of a grid of size ($T \times E \times M$), where a musical sequence consists of T timesteps, a given timestep t includes a maximum of E possible events, and each event contains M modification parameters for capturing details like expressive timing and dynamics.

Figure 5.1 shows an example of a drum machine interface with 16 timesteps and 12 instruments, which can be represented with a Fixed-Grid using $T = 16$ and $E = 12$; the controls for ”Tempo” and ”Shuffle” can each be implemented with a single parameter that applies to the entire sequence. Figure 5.2 demonstrates a different Fixed-Grid design for a drum machine, which includes an option to let users switch between resolutions of T along with three settings for velocity defined by the ”dynamic” parameter.

In addition to the role they play in drum machines, Digital Audio Workstations (DAW's), and other musical devices, Fixed-Grids are common choices for music modeling and generation. Recent examples include MidiNet [109], which generates melodies and chords, and MusicVAE [26] which creates melodies and drum patterns. MIDI-VAE [110] models instrument dynamics in addition to sequences of notes, and GrooVAE [6] includes both instrument dynamics and expressive microtiming. R-VAE [87] and InpaintNet [111] explore finer resolutions as well as ternary divisions on a grid, with R-VAE modeling timesteps as small as 32nd-note triplets.

The main downside of Fixed-Grid representations in the context of machine learning is that it can be difficult to choose an appropriate resolution for T. Too fine a resolution (such as a 1/128th note) results in long and sparse sequences that are difficult to model, while too coarse a resolution (like an 8th note or 16th note) can result in a lossy representation where some notes need to be quantized or discarded [112, 6]. This tradeoff often arises when music is sparse in some places and dense in others, which happens commonly when fast runs or alternate articulations are played on the same instrument.

Event-Based representations

Event-Based representations also have a long history in music generation; they have been used in models based on Markov Chains [113, 114], Recurrent Neural Networks

[115, 116, 117] and Transformers [18, 107]. In contrast to Fixed-Grid representations, which keep track of an event's temporal position by encoding it relative to a specific point on a timeline, Event-Based representations track the passage of time through a discrete vocabulary of *time-shift* events, each of which moves a playhead forward by a specific increment. These increments can be measured in musical durations like 8th or 16th notes, for example to generate jazz improvisations [114] or folk tunes [117], but of particular interest here are a recent series of models of expressive performance that use more fine-grained timespans, with vocabularies allowing time shifts as short as 8 milliseconds. These extended vocabularies of time shifts makes room for models to learn directly from data in formats like MIDI without explicitly modeling tempo and beat.

PerformanceRNN [17] and Music Transformer [18] both take this approach, using Event-Based representations handling time in milliseconds to generate piano performances. REMI [107], replaces milliseconds with beat-based timesteps along with a modifier to handle local tempo variations in an Event-Based representation for pop piano music.

The main downside of Event-Based representations that measure time at high enough resolution for expressive music generation is that in exchange for flexibility, they sacrifice metrical and grouping structures that are connected to the way humans perceive music [118]. Empirical results show that generative models trained with these representations tend to sound less realistic than similarly parameterized models trained with a Fixed-Grid and can have trouble maintaining steady rhythms, particularly over long sequences [107, 6].

5.3 Flexible Grid Representations

To address some of the challenges posed by Fixed-Grid or Event-Based data representations, we introduce a new data representation called a Flexible Grid (visualized in Figure 5.3). Our design for this representation stems from the following question: How can we best encode every musical event in a dataset of expressive performances into fixed-length sequences without needing to quantize or discard any notes?

Avoiding quantization with continuous offsets

As a starting point, we begin with the data representation proposed in chapter 2 for the GrooVAE model (which we treat as a baseline for experiments in section 5.5). This representation, used for modeling expressive drumming with a kit containing 9 drums, encodes drum hits onto a 16th-note grid along with two continuous modification parameters that define, respectively, a velocity v between 0 and 1, and a timing offset o between -0.5 and 0.5, which indicates where between two adjacent metrical positions a note onset occurs. Using the notation from Section 5.2, one measure of drums can be represented by GrooVAE in a Fixed-Grid of size $(T = 16) \times (E = 9) \times (M = 3)$. Because of the continuous offset parameters, the drum hits captured here do not need

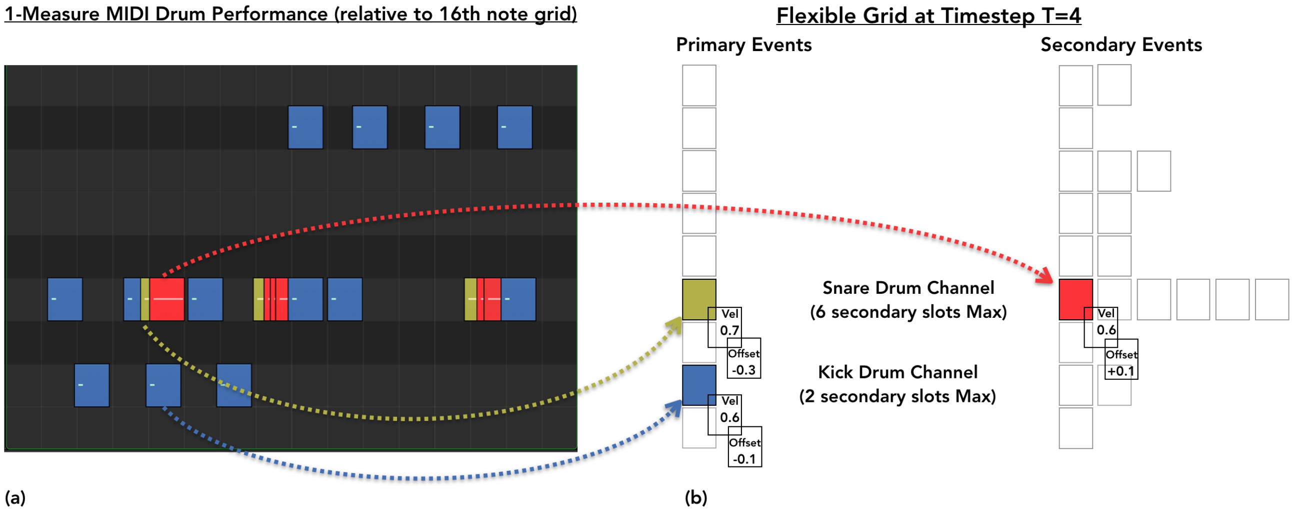


Figure 5.3: (a) One measure of drums from the Groove MIDI Dataset visualized in pianoroll format. In a grid at 16th-note resolution, 9 of the 15 snare drum hits in this measure would be mapped to duplicate slots in a matrix; of these, only 3 notes (colored in yellow) could be kept, and the other 6 (colored in red) would need to be discarded or quantized. **(b)** Mapping drum onset events to slots in our proposed Flexible Grid data representation. Red notes are considered secondary. Each instrument channel (kick, snare, hi-hat, etc.) receives one primary event per 16th note timestep, and space for secondary events is distributed with the minimum number of slots needed to fit the densest passages in the training set. Every event here has two continuous modification parameters for velocity and timing offsets.

to be quantized, so microtiming is preserved at the same resolution it was originally captured. Evidence from several studies indicating that timing fluctuations at the level of individual notes are better explained as deviations from a local tempo rather than as short-term changes in tempo [103, 119], supports this choice of representation using timing offsets rather than tempo changes. Building off of this representation, we use the same modification parameters v and o to accompany each event in a Flexible Grid.

Avoiding skipped notes with secondary events

The Fixed-Grid representation used by GrooVAE breaks down, however, when more then one onset occurs at the same timestep on the same instrument channel. This is a common occurrence whenever a fast musical gesture spans multiple onsets (e.g. a flam, roll, or double stroke on a snare drum).

Figure 5.3**(a)** shows one example of a measure from the Groove MIDI Dataset that leads to this problem: At three different points in this measure, the snare drum channel contains two or more events mapping to one point in time and so cannot be fully captured by the Fixed-Grid at 16th-note resolution. Of 9 snare drum onsets, only the 3 shown in yellow are preserved, while the 6 shown in red are ignored. Whenever we run out of slots in the matrix like this, we need to make a choice about which to keep; in chapter 2 we chose to keep the loudest event when faced with this decision. While the reasons underlying this kind of quantization are not easy to make transparent to users of tools built on these representations, low-level decisions like this can have a far-reaching impact on the ways that tools actually can be used.

One way to avoid skipping notes is to increase the resolution of T from 16th notes to 32nd notes, 64th notes, and so on [6, 87]. This approach, however, does not easily resolve the problem; in the example shown in Figure 5.3, a 32nd-note resolution still misses 4 of the 9 notes in question, and a 64th-note resolution misses two. Moreover, increasing the resolution makes sequences longer and correlations between related positions in the grid less regular. Previous results [6], show that music generation models using Fixed-Grids with too high a resolution are more difficult to train and produce more audible artifacts. A second option for avoiding skipped notes, then, is to switch to a tempo-free Event-Based representation in order to bypass the problem through the use of variable length sequences. This choice, however, comes at the cost of potentially less data-efficient training and generated outputs that may accumulate timing errors over the course of a sequence.

Rather than taking one of the above approaches, we instead observe that the snare drum events in Figure 5.3 can be accommodated into a grid if we allocate three extra slots for snares, increasing the E dimension of our matrix from 9 to 12 so that at each timestep, we have a maximum of 4 snare drum events along with one event for each of the other 8 instruments in the drum kit. This simple change lets us encode this entire measure into a grid of dimension $(T = 16) \times (E = 12) \times (M = 3)$ without any dropped events. Viewed another way, we concatenate our *primary* grid P, of size $(T = 16) \times (E = 9) \times (M = 3)$, with a *secondary* grid S of size $(T = 16) \times (E = 3) \times (M = 3)$. P encodes the blue and yellow notes in Figure 5.3, while S encodes the red notes.

Encoding in this way can provide two advantages over increasing the temporal resolution: first, a smaller and denser matrix gets us to the point where we do not lose any data, and second, the musical events featurized by the secondary matrix S share a common structure that differs from the events in the primary matrix P: all of these events represent musical gestures moving faster than the subdivision of the grid, and they all occur in close proximity to other events on the same channel, which presumably correspond to other onsets produced by the same gesture (e.g. a drumroll). This method of constructing S does not have the undesirable side effect of degrading the rich correlation structure in P (P is left unchanged), which happens when we increase the resolution of T from 16 to 32. Another way to think about why this representation should be beneficial is that, similar to the way in which Fixed-Grid representations make machine learning problems easier by injecting information about metrical position, separating primary and secondary events injects contextual information about musical gestures into the data representation. Taken together, P and S make up a Flexible Grid representation of the drums shown in Figure 5.3.

Applying a Flexible Grid to a dataset

Although we can encode the measure shown in Figure 5.3 into the P and S matrices above, surely there are other measures in our dataset that will not be captured by that encoding, in which we have secondary slots for snare drums, but not for the other 8 drum channels.

If we generalize our method of expanding S, however, we can construct a Flexible Grid that fits every sequence in the data; this can be thought of as making space in S for events that happen as fast as the fastest gestures in our data, but no faster than that. To do this with the Groove MIDI Dataset, we map every drum onset to its closest 16th note timestep, count the number of onset events mapped to each instrument channel (snare drum, kick drum, closed hi-hat, etc.) at every timestep, and then take the maximum value of this quantity for each of the 9 drum instruments that occurs anywhere in the entire dataset. These resulting 9 values E_c (representing the maximum number of possible events for each channel) correspond to the maximum number of times that each instrument in the kit was played within the span surrounding a single timestep.

Table 5.1 shows the result of this computation applied to the Groove MIDI Dataset at 16th-note resolution: we design S to fit one extra ride cymbal, 2 more open hi-hats, 6 additional snares, and so on; by adding a total of 21 extra grid slots per 16th note, we can capture every event in the dataset.

Drum	**Maximum Number of Onsets Within a 1/16 Note Window**
Kick	3
Snare	7
Closed Hi-hat	4
Open Hi-hat	3
Low Tom	3
Mid Tom	3
Hi Tom	3
Crash Cymbal	2
Ride Cymbal	2
Total	30

Table 5.1: Statistics of the Groove MIDI Dataset used to build a Flexible Grid Representation at 16th note resolution.

This approach to constructing primary and secondary grids and encoding a musical sequence into the relevant locations can be summarized with the following sequence of steps:

1. Associate every event in a musical sequence with the closest point in time on the grid.

2. For each input channel c (in our case one of 9 drums in a drum set), count the maximum number of events E_c that have been associated with any single timestep.

3. Set the dimension of E in the primary event matrix P to be equal to the total number of instrument channels, and set the dimension of E in the secondary matrix to $\sum_c (E_c - 1)$.

4. When encoding a sequence, map the first event at time t and channel c into the corresponding position in the matrix P, along with its modification parameters for velocity and timing offset.

5. If there are any more events at time t on channel c, map each of those in temporal order to the corresponding positions in the matrix S, such that subsequent slots will never be filled without first filling all previous slots. If there is no event in P at time t and channel c, then S cannot contain any events at that position.

Considerations for designing Flexible Grids

Of course, the choices of what events to consider as primary depend on the content of the music in the dataset and especially on the types of repetition that take place most often. For example, if our dataset contains many 8th-notes and 8th-note triplets, as pointed out by [87], we may benefit from constructing a primary grid that includes both of those resolutions. Or, to take another example, if our dataset contains many possible pitches (e.g. 88 piano keys), we might want to fit the more common pitches, like those in the current key center, into a primary grid, while leaving slots for out-of-key notes to a secondary grid (e.g. with modification parameters for sharps, flats, octaves, and so on).

While the structure outlined here is perhaps the most straightforward arrangement of a Flexible Grid and could be plugged into drum machine interfaces or off-the-shelf machine learning models, given an appropriate model or musical context, the secondary matrix S could be structured differently, for example as a variable-length sequence in a hybrid setting alongside the fixed matrix P.

5.4 Experiments

Data

To explore Flexible Grids and compare with other representations in the context of machine learning models, we conduct experiments using data from the Groove MIDI Dataset [6]. This data consists of about 14 hours of professional drum performances (recorded by a total of 10 drummers) captured in MIDI format on an electronic drum kit. It was recorded by drummers playing along to a metronome, so we are able to assume a known tempo and downbeat (this is one of the main structural assumptions we need to make; in situations where this information is not captured with the dataset, we would need to automatically infer these quantities using beat-tracking). The drumming in this dataset is representative of typical rhythmic patterns from several styles including jazz, latin, and rock music. For our experimental setup, we divide the dataset into 2-measure segments with a 1-measure sliding window, following the same procedure as in chapter 2. This results in a training set of about 17000 2-measure drum sequences and development and test sets containing about 2200 sequences each.

Data representations for comparison

For experiments, we consider four baseline data representations, keeping machine learning model architectures, hyperparameters, and training procedures the same, while changing the data representation.

Fixed-Grid(16) This baseline corresponds to the data representation used in chapter 2 for generating drums with the GrooVAE model. Here, events for each of the 9 drum categories are encoded using a fixed grid at 16th-note resolution, with continuous modification parameters for each event's velocity and timing offset relative to the nearest 16th note. A 2-measure drum sequence is represented using a grid with dimensions $(T = 32) \times (E = 9) \times (M = 3)$.

Fixed-Grid(32) Here, to add resolution in the time domain, we increase the number of timesteps T from 16 to 32 per measure, so a 2-measure sequence has dimension $(T = 64) \times (E = 9) \times (M = 3)$.

Fixed-Grid(64) This representation further increases the number of timesteps per measure to 64, using a grid of dimension $(T = 128) \times (E = 9) \times (M = 3)$ to represent two measures of drums.

Event-Based For this baseline, we use the Event-Based representation from Oore et al. [17], where MIDI notes are converted into variable-length sequences using a vocabulary V of 9 *Note-on* events, 127 *Time-shift* events from 8-1000ms, and 32 *Set-velocity* events (*Note-off* events are not needed for our percussion dataset). With this data structure, 2-measure sequences are represented by a variable length matrix of size $(T = t) \times (V = 168)$, with the sequence length t taking values up to 300 (the largest number of tokens in this vocabulary needed to represent any 2-measure sequence in the training set). We convert all data to a tempo of 120BPM before any other processing.

Flexible Grid We use a Flexible Grid constructed at 16th-note resolution as described in Section 5.3. The P component of this representation is equivalent to the first baseline, **Fixed-Grid(16)**. The S component is a secondary grid of size $(T = 32) \times (E = 21) \times (M = 3)$. For modeling, we concatenate P and S along the E dimension into a $(T = 32) \times (E = 30) \times (M = 3)$ grid.

Analysis of skipped notes

Because the drumming in the dataset is quite varied, the prevalence of different kinds of gestures also vary depending on the drummer and the musical material. We first extend the same analysis applied to the measure in Figure 5.3 to the entire dataset in order to understand how many notes are quantized or dropped by each data representation.

This measurement aims to give a sense about the scope of the impact a data representation can have when used in models. If a representation drops many events, this effect will always be passed on to any models that use it. If a representation does not drop any notes, we can say that it has the potential to accurately model all the details of the data; of course, the question of evaluating how those models actually perform is left for subsequent modeling experiments.

Music generation with VAE

Next, we explore training a Variational AutoEncoder model to unconditionally generate 2-measure musical parts. In practice, this model can be used for generating new drum loops, interpolating between existing loops, or other applications that motivate research into VAE's for music [26]. While this experiment aims to capture the most general setting for generation in order to best isolate the effects of the data representation, VAE's also include encoders (unlike autoregressive models or Generative Adversarial Networks), which are important for any creative applications that involve conditional modeling based on user input control signals like MIDI scores or rhythmic performance gestures [6].

For our model, we adopt the Recurrent-VAE neural network architecture used in chapter 2. While examining a variety of different models here in conjunction with choices of data representation merits further exploration, we restrict ourselves to one model here to focus on differences between representations. This architecture is convenient because it lends itself well to both fixed and variable-length sequences; we are able to use the same network for all 5 conditions including the Event-Based representation. We follow the same choice of hyperparameters as in chapter 2, except for reducing the value of the VAE regularization parameter β from 0.2 to 0.002 (increasing the weight given to the reconstruction loss component of the objective function), which we found worked better for the baseline model before adopting this change for these experiments.

We train 5 VAE models, one using each of the 4 baseline data representations, as well as one using the proposed Flexible Grid representation. We are interested here in both the perceptual qualities of model outputs (how good do they sound?) as well as in the types of gestures that are present in generated music (do they capture the diversity in gestures, generating drumrolls, flams, and so on?).

As one way of exploring differences with regard to perceptual quality, we conduct an online listening survey with 11 expert drummers, asking each participant to provide pairwise rankings for 15 pairs of generated samples (a total of 165 trials), with pairs drawn randomly from a pool of 128 samples from each model. In choosing their subjective preference for each pair, participants are informed that all samples have been generated by machine learning models, but they are not told anything about the differences between groups or about the specific focus of the study. Before running the survey with our participants, in preliminary comparisons by our research team, we found that two of the baselines (**Fixed-Grid(32)** and **Fixed-Grid(64)**), had a noticeably higher proportion of audible artifacts, so we chose to focus our survey resources on the remaining two baselines (**Fixed-Grid(16)** and **Event-Based**) to obtain a larger sample for the most important comparisons. While the survey aims to capture overall subjective differences between outputs from each model, we do not ask participants about more specific differences in order to avoid introducing biases by directing them to listen for particular details (like the presence or absence of drumrolls).

Reconstruction with VAE

In this experiment, we measure the onset-level reconstruction performance of VAE models trained on each representation, reporting F1-scores. Because a VAE may add or drop notes in reconstruction (it is responsible here for joint generation of both the drum pattern and its expressive timing and dynamics), the alignment between original and reconstructed notes is not known. Given a note n_{si} from a sequence s in the test set and a reconstructed sequence r generated by a VAE, we define n_{si} as having been correctly reconstructed if any note n_{rj} of the same category (e.g. snare drum) is present in r and appears within 20ms of the original note n_{si}. We choose this tolerance of 20ms based on an approximate upper bound of the temporal resolution of human listeners' ability to discriminate sounds, which has been shown to vary from as little as 2ms in some cases to about 20ms in others [36, 37]. Each reconstructed note n_{rj} is only allowed to match one note in the original sequence, to avoid rewarding models that average or quantize note timings. To estimate the best alignment between s and r, we use dynamic time warping for each drum instrument, to match snare drums in s with snare drums in r, and so on.

For this evaluation (in which we are not constrained by a limited number of human listeners to make judgements), we include a second model architecture to broaden the scope of our comparisons: in addition to the Recurrent VAE used for the generation experiment in Section 5.4, we also train a Convolutional VAE using each data representation (except for **Event-Based**, which requires a network capable of processing variable-length inputs). Here, we replace the recurrent networks with convolutional encoders and decoders based on the DCGAN architecture [120], adjusting the numbers of convolutional filters so that each model has approximately the same number of parameters.

Classification

In our final modeling experiment, we compare the different data representations for two classification tasks given a 2-measure sequence: the 10-way classification task of predicting the identity of the drummer, and the 18-way task of predicting a genre as labeled in the Groove MIDI Dataset. We use an MLP neural network model with a single hidden layer for this experiment, again fixing the model architecture and varying the data representation. The hypothesis here is that the features defined by the different ways of representing the same data may be more or less discriminative for categorizing music by performer or genre; for example, drummers may play the same pattern but express it through different stylistic gestures in their playing.

5.5 Results and Discussion

Analysis of skipped notes

Table 5.2 displays the total numbers and percentages of notes that are dropped when we convert from MIDI to each data representation and back, without doing anything else. **Fixed-Grid(16)** drops 6.94% of the events in the dataset, which gives a sense of how much detail is lost in existing models using the representation from chapter 2. Increasing the grid resolution in **Fixed-Grid(32)** and **Fixed-Grid(64)** cuts down this number significantly to 2.85% and 0.92% respectively. The Event-Based representation does much better according to this measurement, only causing distortion in time for 0.1% of the drum hits.

We also find that 58% of the 2-measure sequences used in our modeling experiments have at least one drum hit that is dropped when using the **Fixed-Grid(16)** representation. This means that for 42% of our datapoints, **Fixed-Grid(16)** is sufficient for encoding all the data.

Representation	Total Notes Skipped	Percent of Notes Skipped	Size
Fixed-Grid(16)	24038	6.94%	$32 \times 9 \times 3$
Fixed-Grid(32)	9875	2.85%	$64 \times 9 \times 3$
Fixed-Grid(64)	3210	0.92%	$128 \times 9 \times 3$
Event-Based	348	0.10%	$X \times 168$
Flexible Grid	0	0	$32 \times 30 \times 3$

Table 5.2: Statistics of the counts and percentages of events in the Groove MIDI Dataset training data that would be quantized or dropped by different data representations, before any modeling takes place. Variable length sequences in the Event-Based representation are between 4 and 300 tokens long.

Music generation with VAE

Figure 5.4 shows the results of the listening survey conducted with drummers. Three data representations (**Flexible Grid**, **Fixed-Grid(16)**, and **Event-Based**) were each compared against each other; we show the head-to-head results aggregated across all participants for each comparison.

Results show that both grid-based representations were preferred when compared against the Event-Based one (about 70% of the time). This accords with previous results demonstrating the benefits of beat and tempo-relative representations in music generation [107]. The most likely explanation here, which we experienced when piloting the survey ourselves, is that it can be jarring to listen to short drum loops that do not keep at least a relatively consistent beat; many of the samples generated by the **Event-Based** model exhibit this tendency, whereas the clips from the other two models usually do not. One potential confounding factor that could work against the **Event-Based** model in this comparison is that it is responsible for learning about tempo; we control for

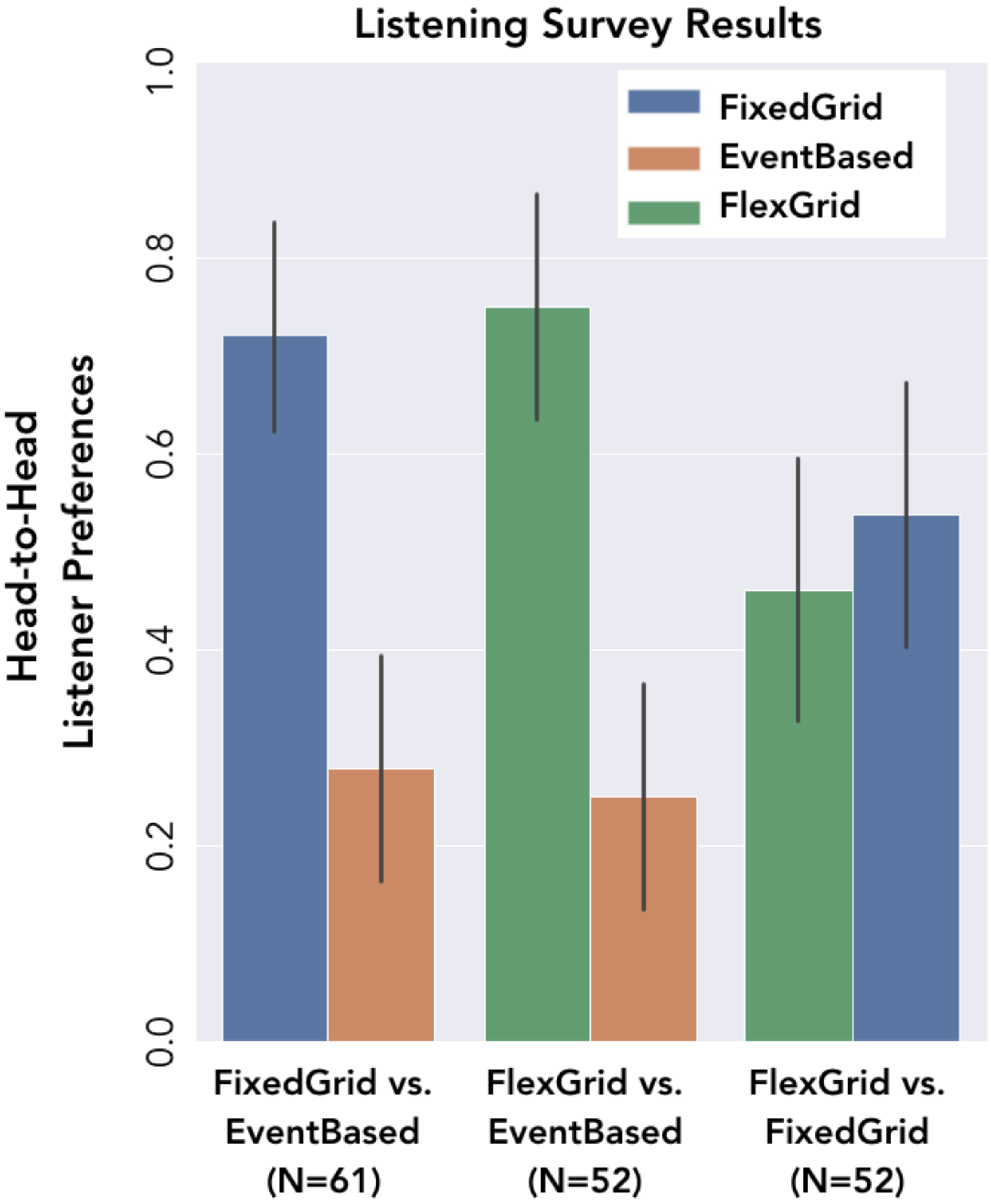
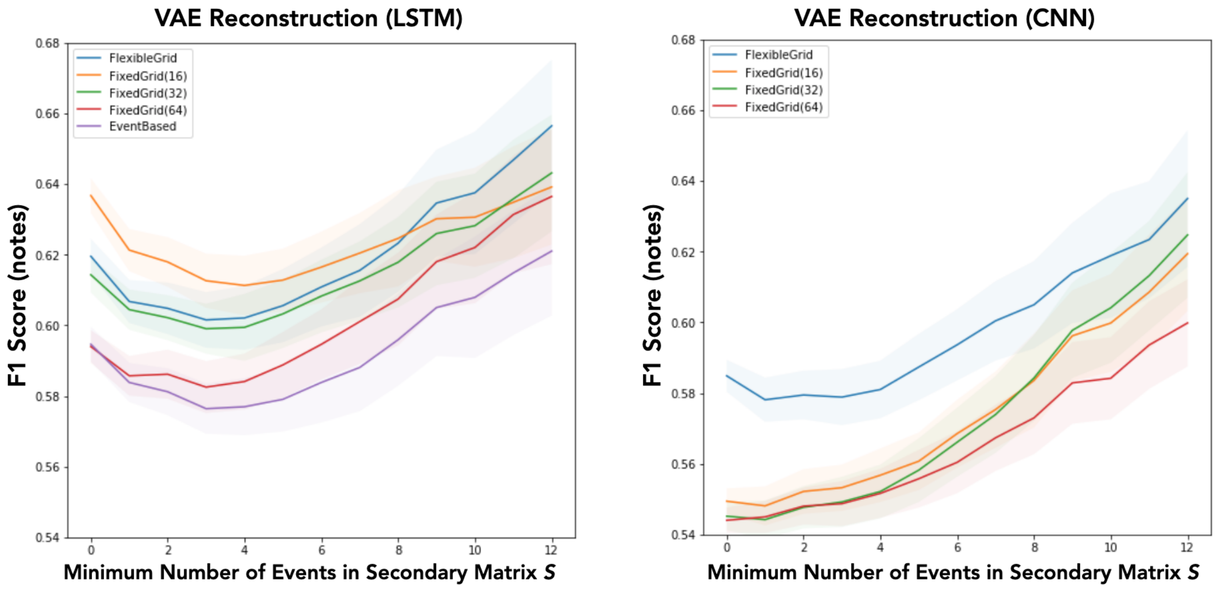


Figure 5.4: Results of a blind head-to-head listening survey. Eleven drummers each participated in 15 trials for this survey, each of them choosing between pairs of two-measure drum loops generated by VAE's trained on each of three data representations.



(a) (b)

Figure 5.5: VAE Reconstruction (F1 scores per onset), plotted against sequences with increasingly more drumrolls and fast gestures. Data are aggregated such that the leftmost point on the line includes all drum sequences, the next point includes all drum sequences that have at least one event captured in the secondary matrix S, and so on.

this factor, however, by converting all sequences to the same tempo (120 BPM) before applying any other pre-processing.

In the third comparison, comparing **Flexible Grid** with the **Fixed-Grid(16)** baseline, we do not find a significant difference between the two groups ($p = 0.34$). Here, the differences between the two models are subtler; the main difference is that **Flexible Grid** is capable of generating a wider variety of gestures like drumrolls and flams (**Event-Based** also offers this capability, but has other drawbacks). In the context of this survey, where drummers were asked to listen to 2-measure loops without any musical context, these gestures, which appear in some samples but not others, did not strongly influence listeners in their choices. Taken together, however, that the **Flexible Grid** model can generate a more diverse set of musical gestures, while at the same time remaining comparable to **Fixed-Grid(16)** and preferable to **Event-Based** in this survey, offers evidence to support Flexible Grids as having combined advantages from each of the two baselines.

Reconstruction with VAE

Figure 5.5 shows the F1-scores for reconstructing sequences in the test set using VAE models trained on each data representation. Each model is trained once and then evaluated across different groupings of the test data. Because not all drumming in the dataset contains the same distribution of gestures, to tease apart differences between representations, we stratify our evaluations here by the number of events that do not fit into the baseline **Fixed-Grid(16)** matrix (and so would be dropped in the VAE's input). If we include the 42% of 2-measure sequences that are fully captured by this baseline, we can see that in the LSTM setting, **Fixed-Grid(16)** performs best according to this metric, with an F1-score of 0.638, compared to 0.620 from **FlexibleGrid**. As we increase the the proportion of sequences with fast gestures in the evaluation, however, **FlexibleGrid** overtakes the baseline here when considering only sequences with 9 or more secondary events. In the CNN setting, however, **FlexibleGrid** performs best across the whole distribution.

These results demonstrate how the impact of the events captured by each data representation are passed on to models trained using each one. Even though **Fixed-Grid(32)**, **Fixed-Grid(64)**, and **Event-Based** all encode more notes than **Fixed-Grid(16)** (as shown in Table 5.2), the corresponding models are not able to learn as well, so the reconstruction metrics are lower.

Classification

Table 5.3 summarizes the results of models trained to classify drummer identities and musical genres using each data representation. We find that while performance for Genre ID is fairly consistent across representations, **FlexibleGrid** performs better for classifying drummer identities, reaching an accuracy of 0.683, more than 3 absolute points better than the next best model at 0.650. This result suggests that encoding expressive music using a Flexible Grid captures some information about the gestures that each drummer uses which can help to discriminate between the different players.

Representation	Drummer Classification Accuracy	Genre Classification Accuracy
Fixed-Grid(16)	0.634 ±0.027	**0.547** ±0.026
Fixed-Grid(32)	0.650 ±0.026	0.544 ±0.026
Fixed-Grid(64)	0.615 ±0.026	0.519 ±0.026
Event-based	N/A	N/A
Flexible Grid	**0.683** ±0.024	0.540 ±0.027

Table 5.3: Accuracy Scores Classifying Drummer Identity with an MLP neural network, with 95% bootstrap confidence intervals. The Event-Based representation is excluded here because the variable-length representation does not enable modeling with a feed-forward classification model.

5.6 Conclusions

Whenever researchers or technology designers work with musical data, we need to pay close attention to the representations we use when converting real world data into formats suitable for computational modeling. Every model that treats music as data must choose *some* representation, and there is a long history of systems and models using different data representations, which we categorize and summarize in Section 5.2.

These choices of data representation are made at an early stage in the series of decisions that shape how music technology is built, designed, deployed, and ultimately put into the hands of creators, and small decisions here can have a large impact down the road.

Previous research suggests that the ways in which creators actually find uses for machine learning-based tools often diverges from the intentions of technology designers [106, 102], and questions around *how* these underlying data representations will ultimately *matter* to music creators [73] may not be thoroughly answered in the near future. Still, as applications based on machine learning become more integrated into the real world creative processes of music producers, composers, and performers of different backgrounds and levels, we can expect that low level choices of representation certainly will matter.

This chapter takes a close look at the relative strengths and weaknesses of different approaches to representing expressive percussion data. We find that Fixed-Grid approaches used in the past have not been able to capture all the rich details of multi-scale musical gestures, while Event-Based representations are often more difficult to train and interact with; in response, we propose Flexible Grid data representations as a balance between these two endpoints. We find that when used for music generation, models trained on Flexible Grids are able to generate music of similar perceptual quality to Fixed-Grids, while at the same time incorporating details of the expressive drumming gestures captured by Event-Based representations. As more datasets and applications are developed around expressive music data (automatic transcription from audio to MIDI offers one path forward), we hope that the underlying motivations and design choices of the data representations explored here will be beneficial in a range of other musical settings.

Ethics and Consent

Ethical approval for this study has been obtained by our institution's Internal Review Board under protocol number: *2019-02-11880*.

Acknowledgements

We thank the editors and the anonymous reviewers at TISMIR for their valuable feedback and suggestions. Thanks also to Jeremy Gordon for help with the survey. The research reported in this article was supported by the Hellman Family Faculty Fund and by resources provided by NVIDIA.

Chapter 6

The Stories Behind the Sounds: Finding Meaning in Creative Musical Interactions with AI[1]

Abstract

Through a series of three studies, this chapter probes the experiences of musicians, producers, and composers as they attempt to introduce machine learning into their creative processes. In chapters 2 through 5, I covered a range of approaches to making music with machine learning. The algorithms, models, and tools discussed along the way are motivated by the needs of particular groups of potential software users, who we can characterize in one way or another based on who we think they are. In designing for these users, I (and others in the field) typically focus on coming up with new technology to fit into existing creative processes. But what happens when we move beyond imaginary personas and into the real world? How do real-world interactions between people and AI music systems play out?[2] In this chapter, I start by taking a step back from specific musical problems like beat-making or orchestration, instead focusing on the experiences of musicians or listeners when we introduce machine learning into creative processes.

My approach toward the studies in this chapter acknowledges that every creative process is different. Some people write music with an instrument, some write with a computer, some use pen and paper; sometimes we create to meet a deadline, sometimes we create because we feel inspired or emotional; sometimes we create alone, sometimes we work together. We all change from moment to moment and year to year as we go through different experiences and face different situations. Might our needs and experiences in working with musical AI be similarly individualized?

[1] The material is this chapter draws on work done in collaboration with Noura Howell, Wesley Deng, Julia Park, Yangyang Yang, Carmine-Emanuele Cella, David Bamman, and Kimiko Ryokai.

[2] I emphasize the term **AI** in this chapter, rather than the term machine learning, in order to better reflect the norms of the communities of people working with these technologies as part of their creative practices.

In part because of the always-changing contexts in which people create music and art, studying human interactions with AI in situated creative environments is hard. Controlled studies "in the lab" might separate creators from their usual processes in ways that color their experiences, making it difficult to isolate the effects of new AI technology [106]. Participants brought in to try out prototypes or learn how to use AI-based creative tools for the first time might find their learning curves to be steep; it can take a long time to start to understand how AI works or how to use it. And finally, participants might not feel very invested in the outcomes of their (often unfamiliar) interactions with AI. The research in this chapter begins with the following question: what are experiences and interactions like for people who *have a reason to be emotionally invested* in music created with AI? I approach this question from different angles in each of the sections that follow:

- Section 6.1 uses first-person design research methods to probe the experiences of a group of people (who are not necessarily musicians) *listening* to individually customized music that uses samples and stories from meaningful moments in their own lives.
- Section 6.2 describes my own firsthand experience producing a song together with a group of 4 people that ended up as the winning entry submitted to the 2021 AI Song Contest, an international contest exploring the potential uses of AI for songwriting.
- Section 6.3 builds on my findings from 6.1 and 6.2 through case studies with two musicians, one professional and one amateur, working with AI to manipulate sounds from their lives in order to create musical materials (new samples, loops, or digital instruments) to compose with.

6.1 Listening to AI Music: Perspectives on Sampling and Remixing with Personally Meanginful Sounds

6.1.1 Introduction

What personally meaningful sounds do we cherish in life? Loved ones' giggles, or unique creaks of the door in a childhood home. Energetic yelling at the neighborhood market, or the timbre of a beloved instrument. Ambient environmental sounds, the voices of loved ones, and other sounds from our varied everyday lives may intrigue, energize, or calm us. Sound can make us appreciate what we have or reflect on what we have lost. Such ″personal sounds″ are ubiquitous and rich with emotional meaning, yet they are fleeting and ephemeral. How might we keep and savor these sounds?

For musicians, one way to hold onto personally meaningful sounds is to reuse them as creative material through *sampling* and *remixing*. Artists and producers use samples to convey all kinds of emotions, from the pain and anger of Pharrell Williams sampling police brutality protests [121] to the lighthearted nostalgia of singer Billie Eilish and producer Finneas sampling the sound of crosswalk signals from a trip together to Australia [122].

Reinterpreting recognizable and meaningful source material can offer an immediate point of connection that serves to ground a new composition in a particular time, place, or emotion [123]. Because of the way in which they bring familiar and recognizable elements into otherwise unfamiliar music, sampling and remixing present a promising entry point from which to explore, with the participants in this study, the very unfamiliar experience of interacting musically with AI. By establishing an underlying connection between AI-generated music and specific sounds or memories that are already meaningful to the listener, sampling offers a way to raise the emotional stakes for research participants who might not otherwise approach AI-generated music as focused and active listeners. At the same time, the process of collecting and sharing personal sound collections *to be remixed* using AI is itself an opportunity to surface design implications for creative human-AI interaction; much like musicians might guide drum generation models by tapping rhythms along to a click track, participants here can guide model outputs through the choice of samples that they input.

Focusing on emotional experiences with sound, this study asks: How might it feel to collect, share, remix, and appreciate emotionally meaningful sounds from our lives? What would a remix of personal audio recordings sound like, how might such a remix be created, what musical decisions would this entail, and how might such remixes be experienced by the people whose sounds were modified? What parts of the process could be delegated to AI, and what parts could not?

6.1.2 Study Design

Today's methods for creating music with AI are not yet at the point where they can automatically generate entire compositions that listeners are likely to be interested in. A few services like Amper [124] and Jukedeck [125] use AI to generate background music for use in videos, podcasts, or social media, but there is little evidence to suggest that listeners find this generated music interesting or meaningful outside of those functional contexts. As I argue throughout this dissertation, the potentially more promising applications of AI toward music production and creation involve significant human interaction.

For this study, rather than working within the limitations of current AI systems or building an entirely automated system for generating customized music based on samples, I instead explore a speculative future in which AI is already able to automatically compose customized music for participants. I take inspiration from Wizard of Oz prototypes, in which a human plays the role of a machine while interacting with an unsuspecting human participant [126]. Because of the vulnerability involved in sharing emotionally sensitive stories and audio source material, this study does not employ any deception; instead, to encourage honesty and openness, I employ first-person research methods, serving as one of five participants myself along with four other co-authors.

To set up this scenario, I play the role of the AI (drawing on my own experience working as a composer for clients in film and advertising). I create and deliver customized remixes to participants using samples they provide, while asking participants at dedicated moments through the process to reflect on how they would feel if they were interacting with an AI system as their remixer rather than with me. Together with four other researchers with diverse cultural and professional backgrounds, I study the design space of collecting, sharing, remixing, and reinterpreting personal sounds, critically probing how listeners might experience handing their personal sounds to an algorithm to be remixed. This study contributes: **(1)** several diverse perspectives envisioning a human-centered future for musical human-AI collaboration, and **(2)** nuanced experiential insights on personal sound remix design to open a broader design space for emotional meaning-making, reflection, and remembrance.

6.1.3 Related Work

Personalized Music and Sound Generation with AI

This study is motivated by potential futures imagined by two approaches to research in machine learning and HCI. The first of these looks at AI-generated content from the listener's perspective; e.g., full-length songs composed by OpenAI's Jukebox [1], automatically generated sound effects for films [127], or music for video games [128]. The second approach (to which earlier chapters in this dissertation subscribe) engages with AI from the creator's perspective as another tool in a digital toolbox [129, 73, 130]. Both of these strands of research are moving in recent years toward increasingly personalized, customized content. For listeners, this could be individually tailored jazz improvisations created by Bebopnet [131]. For creators, this could take the form of flexible AI systems that condition their outputs on different kinds of user guidance, as discussed in Chapter 4. The growing scope of personalized sound creation with AI motivates my interest in probing the perceptions of both listeners and creators by studying a scenario with higher emotional stakes, using sounds sampled directly from meaningful moments in listeners' lives.

Design Research using Sound for Reflection and Remembrance

Design research has explored remembrance and personal meaning-making with various collections of personal data [132, 133, 134, 135, 136]. Ryokai et al. explore capturing, cherishing, and reflecting on laughter sounds from everyday life with tangible and visual designs [137, 136, 138]. Olo Radio [139] and Olly [140] allow users to explore their personal archives of music listened to via streaming services. FamilySong shares songs across distance for internationally distributed families [141]. The Affective Diary [142, 143] is an early example of the now many designs that use biodata to prompt emotional reflection, sometimes with sound (e.g., [144, 145]). As voice recognition and sound event detection (e.g., from Amazon [146] or Google [147]) make always-on audio recording more prevalent, this work anticipates growing personal audio archives and explores how design might support engagement with the evocative potential of personal sound recordings. In contrast to ubiquitous computing's typical emphasis on efficiency [148], these designs invite slow, emotional, reflective engagements with personal data [149, 150], with careful attention to emotional experiences of reflection and remembrance with personal data. This chapter takes inspiration from these reflective approaches in similarly centering emotional experiences.

Human-AI Collaboration

Beyond the specific context of music creation, user acceptance of input from AI more generally often depends on a number factors, including the nature of the AI's task [151, 152], various notions of interpretability [153], and users' mental models of how AI works [154, 155]. Adoption of AI is often resisted by both domain experts and general users in high stakes scenarios like medical diagnosis [156, 157, 158] or financial risk assessment [159]. Although AI music generation is usually not considered a high stakes scenario, this study aims to make the stakes higher than they have been in the past through the intentional use of personally meaningful and potentially sensitive sound samples, inviting engagement with tensions that might otherwise have not yet risen to the surface.

Emotions and Subjectivity in HCI

This study takes a subjective, emotional approach in exploring human interactions with AI. Within disciplines like HCI and AI, emotional experience is often denigrated as an inferior way of knowing, perpetuating an emphasis on rationality and objectivity. This emphasis is problematic because what counts as rational and objective often reifies colonial, gendered hierarchies that (intentionally or not) delineate Western white men as civilized, rational, and objective, and everyone else as too uncivilized, irrational, and emotional to generate knowledge. Within HCI, Haraway's foundational work on situated knowledge [160], Bardzell's agenda for Feminist HCI [161], and D'Ignazio and Klein's book on data feminism [162] have all argued for the importance of situated, emotional, embodied ways of knowing.

6.1.4 Process

Four colleagues and I chose to study ourselves through first-person research with a four-step process:

1. **Collecting:** The five of us each selected 3-5 sound recordings of "life-defining" moments, which entailed either finding and choosing an audio recording, or reflecting on the absence of a desired recording (because many of us do not keep audio collections).

2. **Sharing:** We listened to the sounds as a group before explaining their personal, emotional significance to each other.

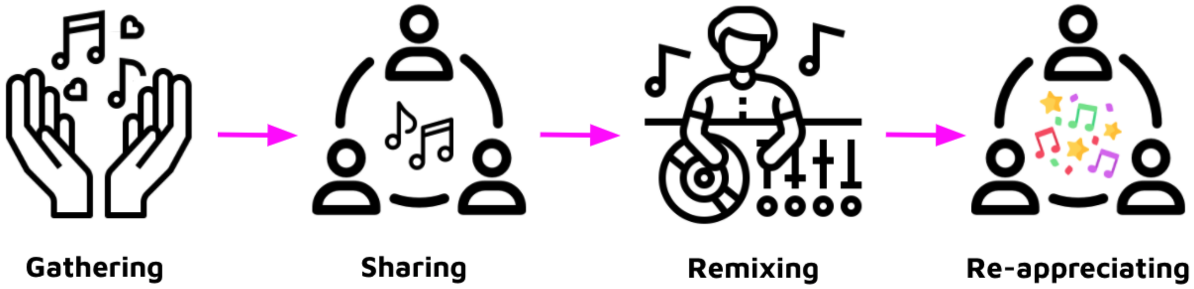


Figure 6.1: Study process: 1. Collecting personal sounds, 2. Sharing the sounds and the stories behind them, 3. Remixing the sounds (with a human musician standing in place of a speculative AI remixer), 4. Listening to the remixes with the group.

3. **Remixing:** I modified and arranged the sounds into new musical or soundscape ″remixes″ for each participant, including myself. Each remix for ″Author A″ used sounds only from ″Author A.″

4. **Reinterpreting:** We each listened to our respective remixes individually, then listened to them as a group, and finally shared our emotional responses to the remixes with the group. As described in Findings and Discussion, these remixes presented sonic, artistic, and emotional reinterpretations of the original sound recordings and their personal significance, with sometimes pleasing, sometimes upsetting, and often surprising results. At dedicated checkpoints throughout our process, we individually logged our emotional experiences, and during meetings we helped one another articulate and further reflect on the emotional meaning of the sounds and remixes. These journal entries and resulting meeting transcriptions were qualitatively analyzed for emergent themes [163].

Engaging first-person methods enabled direct, firsthand experience exploring emotionally sensitive territory. Kumi described the sound of her firstborn's first breaths and wished she could have had an audio recording of this; June shared a recording of her dog shortly before he passed; Natalie's sounds touched complicated memories of a person in her life. While it would have been possible to hold a similar study with external participants by soliciting their recordings and presenting them with remixes, we chose not to do this in part because we wanted richer firsthand knowledge of what turned out to be surprisingly sensitive experiences. Moreover, rather than asking participants to undertake vulnerable emotional work in an unfamiliar setting (affective demands on research participants have been rightfully critiqued in HCI [164, 165]), our familiarity with each other, together with shared authorship in this study, made this engagement feel safer and more equitable.

Team Background

In accordance with the reflexivity of first person methods, we described our professional and cultural backgrounds alongside the personal sounds we shared. Pseudonyms are used for the co-authors other than myself.

- **Jon** is a PhD student in the U.S. researching AI and HCI applied to music. His background includes training in music production as well as experience playing music as an independent artist and contributing composition and sound design to films and ads. Jon shared a recording of himself playing guitar in high school and another recording of a performance from his West African drumming class, which was a formative experience in shaping his thinking about music and sound.

- **Kumi** is a university professor in the field of HCI. She grew up in Japan but spent more than half of her life in the US. She is married and has two young children. Kumi shared a sound of her children with their grandmother giggling and singing when they were together prior to COVID-19 pandemic.

- **Natalie** grew up attending 12 different schools in 5 U.S. states; her sound selections often reflected nostalgia for distant places and times. While navigating frequent changes, a consistent thread has been taking clarinet lessons for many years and practicing often. She also uses field recordings for making music as a hobby. She shared sounds of playing clarinet, of someone she knows playing a familiar song, crickets in the South, and the train in a city where she formerly lived.

- **June** is a graduate student studying emerging technology design in the U.S. Her background is a combination of architectural design, engineering, and product design. She shared the sounds of barks and whines of her terminally ill dog greeting her with unbridled joy when she came home one day, clicks and clacks of her favorite pastime, competitive video gaming, and the creak of a door opening to indicate that her partner was done with his work for the day.

- **Howard** is an aspiring researcher in HCI and AI. He grew up in China and moved to the U.S. five years ago pursuing a career as an academic researcher. Being busy with the curriculum, he flies back to his home country only once or twice a year visiting his family and friends. He shared sounds of his cousin performing an imperfect piece of piano in front of the entire family during the Chinese Traditional Spring festival, clips from his high school graduation ceremony, and a jazz band performing in a cafe where he spent a lot of time with his friends before the pandemic.

Remixing Process

Drawing from my own professional musical experiences, I approached remixing for this study as I would when collaborating with another artist or a client: First, I interviewed the ″sound owner″ - the person who had shared their personal sounds - to learn the story behind the sound and get a general sense of their aesthetic tastes. Next, I chose an artistic direction, including decisions about length, timbre, mood, and whether to deliver a short sound-snippet or a longer piece of music. This included finding one or two ″reference tracks″ for inspiration. In an iterative process, I listened closely to the personal sound recordings, looking for moments that stood out because they either captured an intended feeling or because they had unique and memorable sonic qualities. Finally, I put together the remixes by transforming and combining those fragments. The technical steps involved in these transformations ranged from simply adding effects like reverb, echo, or a filter, to building full musical arrangements by organizing, repeating, and mixing sounds into a larger piece. Each participant received a customized set of 5-7 sounds made using their materials, starting with simple transformations and ending with full pieces of music around 1 minute long.

6.1.5 Findings

Searching for personal sounds as a reflective process

From the perspective of the "recipient" of a personal remix, selecting personally meaningful sounds to be remixed was a journey in and of itself. Instead of going out to record new sounds, most of us first reflected on what "personally meaningful sounds" might be for us; this led us to look for sounds from our past. In the process, the act of focusing on sounds (as opposed to images or videos) challenged us to search our memories in a different way than we are used to. Most of us ended up thinking of sounds associated with particular times and places in our lives, like a childhood music recital, the birth of a child, or the sound of crickets chirping outside during summer in a childhood home. When we did have recordings tied to major life events, we felt grateful. For example, June had intentionally recorded the a video of her dog's happy footsteps shortly before he passed away as a keepsake; she extracted the audio from this video.

Some sounds were difficult or impossible to obtain. For Kumi, the sound of her newborn's first breath came to mind, a sound vivid in memory yet one for which she has no recording. When asked to reflect on how she felt about not having the recording, Kumi responded, *"I have mixed feelings. Ultimately, I am OK with not having the recording of my baby's first breath. I still clearly remember just how viscerally precious that sound was to me. Even if I had an audio recording of it, I am not sure if the audio file would have captured the heartfelt aspects because it was such a high-pitched yet super quiet kind of sound. But at the same time, not having the recording of the breath makes me think about how I might go about choosing to record certain precious sounds in my life in the future, or really try to pay attention to the quality of the sounds I experience because **I may not get to hear it again**."*

The seemingly simple exercise of collecting "meaningful sounds" in our lives made us reflect on how we remember events. For some, the prompt to gather recordings invited renewed attention to the sounds of everyday life and appreciative listening to sounds that are typically ignored. June shared the sound of the door from her partner's office as he opened it. *"I like the sound because I know that's when I could leave my computer, go out, and chat with him."* After finishing this exercise, Howard was inspired to start a practice of deliberately collecting potentially meaningful sounds from his daily life, which to him now feels similar to his practice of taking photos: *"When I'm chatting with my family, especially those who I video chat with less frequently, I will now sometimes open my voice memo app and record part of the conversation."*

Stories give significance to sounds

As a group, we first shared our personal sounds without explaining them in order to probe the group's initial reactions to the audio recordings themselves. Most of us found ourselves trying to guess the context for each sound: was it water, footsteps, was it outdoors? The sounds alone did not prompt much emotional engagement for listeners at first. After learning the context and personal significance behind a sound, however, our listening experiences often changed dramatically, with the sounds seeming to *"come to life"*.

For me, this group listening exercise felt like a unique way to get to know each other better. I wrote: *"Hearing others' sounds was a connecting experience. I tried to make sense of what I was listening to. I imagined what the sounds were about, and I* ***pictured the space*** *where the recordings were happening.* ***I came up with my own version of what the story might be****. After hearing the stories behind the sounds, I felt like I was* ***getting to know the person and see the world from their perspective****. I was engaged in listening to the stories more than I might have been otherwise - I wanted to know what it was,* ***why they chose that sound****."*

Sharing stories along with our sounds also changed the way that some of us felt about the act of listening itself. I wrote: *"This was different from my previous experiences of focused group listening with musicians in a workshop or a class. That kind of listening can be overly technical and analytical sometimes, which distracts from getting to the more important matter of understanding how a sound makes us feel. With this group, everybody had a unique perspective on what they were listening for and sharing, not in a technical way, but in terms of the feeling. I was surprised, in a good way, by the variety of things that people listen for."*

For most of us, in our roles both as "sound owners" and as listeners, the backstories felt equally or more important than the sounds themselves. We were not aware of this though until the sounds and stories were revealed to us in this order. This suggests that emotional connections between sounds and stories (real or imagined) can play a more influential role than we would expect.

Remixes Open Personal Stories for Reinterpretation

June spent about 30 minutes explaining to me the significance behind the recording of her dog, Romie. After hearing the remix I created for her, she found herself feeling happy because the remix seemed to embody her dog's *"goofy"* personality, turning what was a sad clip into a joyous one. June had initially found it hard to listen to the original recording that she shared because it reminded her of her loss. The remix, on the other hand, felt more *"listenable"* to her, because it offered a new way to revisit memories of Romie.

In contrast, Natalie had a negative first reaction to her remix, which included a sample of someone in her life playing a familiar song. *"When I first heard Jon's remix, I felt annoyed and disappointed but not upset. It felt like the remix made sense musically, but* ***missed the point emotionally****. The remix seemed to emphasize the song itself... the musical style was* ***trampling my own****. It also made me realize that* ***I hadn't explained*** *the [connection]* *between me and the person playing the song, so how could Jon have known?"*

Natalie's perspective on the remix she received was also shaped by the group dynamics, the other listeners, and the other remixes she heard afterwards: *"I only felt upset after June shared her remix with the group. Although the sounds of her puppy's last days had been poignant and sad, the remix found a sensitive, respectful way to cherish the good of that moment, and felt to me like a commemorative celebration of her puppy's life. Then* ***as the group listened to my remix, I held back tears****. Having such a strong reaction seems a bit silly in hindsight, but in the moment, I was struck by June's positive experience with her sound and* ***wished my remix could have been a positive experience for me****."* Even though the remix itself was hers to keep, Natalie's experience of listening to it together with the group for the first time remained ephemeral - this "first listen" was a one time opportunity that couldn't be repeated again with the same piece of music. Natalie's experience speaks to the power and impact of hearing sounds for the first time as well as to the strong effect that social environments can have on the way we experience music.

Importance of Trust

Sound owners often felt vulnerable when sharing sounds with the group. Entering the meeting to share the story behind the clip of her dog, June felt apprehensive about how I (in my role as remixer) might reinterpret her story. She wrote, *"Would he completely miss the meaning of the clip to me? Or would I gain a new level of euphoria from listening to his work?"* During the meeting, despite feeling anxious, she remembers speaking fast, desperately trying to cram as many details as possible about her dog into the call. She felt she needed me to understand why her dog was so special and important, and why the clip simultaneously *"broke"* and *"lifted"* her. She worried that if I didn't see the whole picture, it might cheapen the memory and devalue the bond between her and her dog. There was something powerful about using sounds to share something precious to her because a verbal explanation was not enough.

June began to feel more at ease when I showed empathy and made it clear that I, in my role as the remixer, was listening to her. She felt that I was invested in *"doing the clip justice."* For June, it was frightening to *"relinquish ownership of the sound."* She describes, *"I went into the call knowing that the expert was skilled in this area, and once I realized he was an empathetic guy, I felt much more trust than I would have with someone driven by only their artistic ego."*

After listening to her personal remix, June wrote: *"I suddenly broke out into a big smile because the music felt like the artist had tried to embody the feeling of the friendship between Romie and me, as if he really cared – he tried to interpret what I had shared with him. And the fact that someone listened so attentively and cared at all – this forced me to examine the clip as a celebratory experience rather than a solemn one."* Even though the remix presented an emotional interpretation that didn't quite match how June initially felt about the recording of her dog, a sense of trust allowed her to appreciate the new interpretation instead of being left feeling misunderstood.

Natalie, on the other hand, had not been given as much space to explain her personal samples to me in detail; she didn't go through the same trust-building process that June did to reassure her that her remixer would care about her perspective. This contributed to her negative initial reaction to her remix. For Natalie, discussing the remix with the group in a supportive environment after this first listen marked a turning point. *"At this point, I felt a flood of gratitude and joy. I had shared my story and negative reaction with the group. Jon had listened to me and explained his thinking making the remix, and I was struck by how much thought and sensitivity he had put into it. Even if it missed the mark for me in terms of the sound itself, his thoughtful care felt like such a gift. I felt compelled to send a thank you email after the group meeting."* While my remix missed the mark for Natalie, hearing about my process of making the remix and learning how I had interpreted her samples shaped her next listen into something different. Natalie wrote afterwards: *"During those few weeks [collecting and sharing sounds], I had been feeling very nostalgic for times gone by and places I couldn't return to.* ***Hearing the sounds from different aspects of my life stitched together into the remix provided a sort of synthesis and healing****."* After learning more about the intentions behind the remix, Natalie interpreted the music very differently; this time it left her with a positive emotional reaction.

From my perspective playing the role of remixer, the need for trust came with a sense of personal responsibility. After remixing the personal sounds for all five members of the group including myself, I wrote: *"I found the whole process to be a really powerful composition exercise. Even though I have some experience playing this role through collaborating on projects with other musicians or artists, I felt* ***a heightened sense of responsibility*** *toward the sound owners here. Rather than thinking more broadly about who the audience for each piece might be, I was focused on the one person who shared the original sounds, because the emotional stakes were so much higher for that person. I did think a bit about whether and how other listeners in the group would respond - but for that secondary audience, I was less concerned with their emotional responses. Because of these high emotional stakes, I spent more of my composition time trying to* ***capture the stories at a high level*** *and less time executing the details of any individual piece than I might have otherwise. While this composition process was engaging for me, it was also challenging. In particular, when my understanding of the story behind a certain sound was less thorough, I felt like I was making guesses and leaving things to chance - giving myself the opportunity for a round of revisions would have been a nice addition to the process."*

6.1.6 Discussion

Perspectives on Delegating Creative Work to AI

Is it possible to trust an algorithm with remixing our personal material, with reinterpreting our precious personal stories? Lustig et al. highlight the increasing role of "Algorithmic authority" in society to "interpret, decide, and manage" [166] complicated situations that arise throughout everyday life. Howell et al. found that, when faced with a data-driven algorithmic interpretation of their own feelings, participants sometimes granted the display a concerning degree of authority, "trusting" it more than their own felt sense [167]. What are the benefits and risks of trusting or not trusting an algorithm to work in emotionally sensitive spaces?

Kumi's Perspective

Kumi is open to having an AI-based system "surprise" her once in a while with sounds from her life, much like the way Apple iPhoto or Google Photos provides "On This Day" (time based) or "This Place" (location based) memories automatically selected from the images and videos on a phone: *"When I see these automatically generated 'memories' on my phone, I am not expecting them to be perfect or even getting it right. I don't expect AI to 'understand' my life. But it is more about treating them like a gift of opportunity to reflect on things that would otherwise go unnoticed."* For Kumi, engaging in this exercise of attending to personal sounds made her realize *"just how little we pay attention to the sonic qualities of our lived lives."* Kumi explains that *"Currently, there isn't any system that suggests to us, 'Here's a personal remix of sonic experience based on where you were or who you were with. It may be imperfect and we might ignore it at first, but once in a while, there might be an opportunity for us to say, 'Hey, I appreciate this sound because I would not have noticed it otherwise, given the myriad of things going on.'"* The AI generated remix of personal sounds might also be experienced much like the way we interact with traditional radio stations. *"If it ends up playing something that I do not like, I would simply switch the channel. With a human artist, we might be too polite to tell them I don't like what you made for me, but with a non-human AI musician, we might be more open to, or even empowered to speak up what we like or not like, and what we hope to see next, and be open to multiple iterations."*

Howard's Perspective

Howard envisions a cloud-based system that could provide him with a palette of sounds that might be interesting to him to remix on his own. This could be based on all the sounds he deliberately records from different apps and devices. Howard would like to be informed about the specific model the AI musician might use and the dataset the model was trained on, possibly through documenting methods like Model Cards [168] and Datasheets for Datasets [169]. With a sufficient level of transparency, Howard may be able to imagine having AI involved in an entire music creation pipeline.

Natalie's Perspective

Natalie believes it would be extremely difficult to provide a positive emotional experience with AI. Even with emotion-tagging audio, even with genre-matching to produce something in a style favored by the listener, and even with AI-generated music, interpersonal trust and relational care are what made the process meaningful for Natalie. Rather than attempting to use AI to generate music instead of musicians, AI can help in other ways. Adding human-AI collaborative features to existing digital audio workstations (DAWs) can provide greater creative possibilities for musicians to work with while remaining sensitive to emotional nuances. AI can help narrow the search space while leaving final decisions for humans, musician and listener. AI might even help listeners find musicians based on musical or emotional preferences, or other characteristics. For Natalie, rather than attempting to use AI to supplant the already precarious labor of musicians, it seems more equitable and just to instead use AI to help provide paid gigs for musicians, in the process enabling more non-musicians to engage the expressive potential of remixing personal sounds.

My Perspective

For me, coming out of this study, I saw the potential for human-AI co-creation with personal sounds as a tool for creative inspiration: *"Hearing something that is uniquely yours suddenly take shape in a more fully realized form than you knew was possible - that can be inspirational and encouraging."* I imagined a future where AI sometimes serves the purpose of emphasizing sounds worth listening to (even if the AI often misses the mark). At the same time though, after seeing firsthand examples of the surprisingly powerful emotional impact that personalizing and remixing sound can have, I started to picture a scarier future in which AI-generated music is deployed in a manipulative way - for example, to play on consumers' emotions in order to encourage them to spend money.

Privacy and ownership

Through the process of handing over life-defining sounds (whether to a human or an AI) to be remixed and re-contextualized, this study offers a look into privacy considerations that can arise when designing personalized sound and music. While work in media studies, musicology, and law has examined ethics, cultural practices, and legal implications of sampling and remixing in music (e.g. [170, 171, 172]), the sampling discussed in this literature usually takes place between one artist and one sound owner (e.g., Public Enemy sampling James Brown). Personalized music created with the help of AI, however, opens the door to the possibility of sampling on a much larger scale, drawing from any individual's sounds that have been recorded and made available.

Considering a future in which some form of sampling can be automated raised questions about what constitutes ethical use of sounds recorded in everyday life. Sounds are all around us, able to be recorded, but ownership can be unclear. Most sound owners felt uncomfortable with the idea of sharing precious sounds outside the research group without permission–and yet, some of the personal sounds shared within the group were recorded with others present or even in public spaces. What if a recording captured by one person contains another person's precious moment?

Consent, intention, and sensitivity to cultural histories emerged as common themes when reflecting on the experience of allowing our sounds to be remixed. I felt uncomfortable with the idea of sampling one of the sounds I had shared with the group, because I knew that a person whose voice was captured in that clip would not want to be remixed: "*It feels wrong to change this recording from its original form. I know he would disapprove.*" Natalie shared having experienced a similar feeling while walking with a friend who expressed wanting to record the sound of people singing during a baptism in a river as they passed by - "*You can't use that!*" Kumi noticed one particular segment in my remix for her in which the sounds of her Japanese family were changed to sound different from their original cultural context. She felt surprised (but in her case, not uncomfortable), describing the sounds as *"Church like! Western European, unexpected and really different from what it is!"*

6.1.7 Conclusions

This study contributed first-person design research exploring a process of collecting, remixing, sharing, and reinterpreting personal sounds. We found that while personally meaningful sounds were expressive emotional materials to work with, they also presented chances for misunderstanding and discomfort. Interpersonal trust and communication of stories behind sounds played an important role in shaping experiences with these sensitive materials. These findings contributed to the approach that I went on to take in the studies on human-AI co-creation presented in Section 6.2 and 6.3, in which I served as a mediator between musicians and AI while facilitating musical collaboration.

6.2 Making a Pop Song with AI

AI Song Contest 2021

The AI Song Contest is an international event, which teams from around in the world can enter by submitting a 4 minute song created using AI in some part of the process. The second edition of this contest was held in 2021: 38 teams participated, with winners chosen through a combination of an online popular vote and a jury of 8 musicians and AI experts headlined by singer Imogen Heap. Songs were judged based on both their quality and on their process, with particular emphasis on how they made creative use of AI.

My approach to and experience in making music using AI for this contest was shaped significantly by the study presented in Section 6.1; I completed that study in February 2021 and began organizing a group to participate in the contest in March 2021. Here, I built on ideas related to "Personal Sounds", but this time with a focus on a group of musicians rather than listeners. As co-creators of a song, each participant on my team played a direct role in the song creation process, as opposed to passively allowing someone else to use their materials.

The rest of this section contains an expanded version of the "Process Document" that I wrote and submitted to the contest's jury, detailing how my collaborators and I created the song "Listen to your Body Choir," which was eventually chosen as the winning entry for the 2021 contest.[3] The song can be streamed at the link in the footnote below. [4]

[3] https://www.aisongcontest.com/blog/ai-song-contest-2021-winner

[4] https://soundcloud.com/user-703675253/listen-to-your-body-choir

6.2.1 Team and Roles

Team Members

- Jon Gillick is a researcher working at the intersection of Artificial Intelligence, Human Computer Interaction, and Music. He has a background in both music production and computer science.
- Max Savage is an audio and video producer with over a decade of experience in creative production. He produces, mixes, and masters music for a variety of artists and is the go-to producer for a number of singer-songwriters and bands that he works with regularly.
- Matt Sims holds a PhD in English Literature and researches how artificial intelligence can be used to understand and explore the role of narrative across different genres and mediums.
- Brodie Jenkins is a recording artist, singer, and songwriter.

Team Formation

The inception of this team began when I, having worked on AI music research for the last few years (after spending the preceding years working in music production), wanted to bring things full circle by seeing if it would be possible to connect my research more closely and personally with my music creation practice, which brings me a lot of joy. I recruited Max, a former music-school colleague, and Matt, a research collaborator who also shares a passion for music production. Finally, Max recruited Brodie to round out the band as lead singer.

None of us knew quite what to expect, but everybody jumped in with an open mind, a sense of humor, and a piqued curiosity. Max and Brodie were excited at the opportunity to work on a strange, unique artistic project; both had seen bits of AI-generated art popping up in the media from time to time, but were not very familiar with the details of AI. Max was also intrigued at the prospect of learning about some new tools and plugins in the process. I felt a bit uncertain about what I had just talked the group into doing, but I was excited at the chance to work with some talented and creative people.

Our Roles and the Role of AI

We assembled a team with skills and backgrounds that we felt would be complementary: A producer to craft the overall structure of the track (Max), a lead singer (Brodie), a producer with expertise on AI tools for music generation (Jon), and an expert on natural language processing and narrative (Matt). Each member used their unique skill set to focus on a specific aspect of the song creation process, but there was also constant communication, coordination, and input between the team members to ensure that we maintained a unifying vision and that our individual expertise didn't produce a fragmented aesthetic. We set up weekly zoom calls, a shared Dropbox folder, and tons of email chains to share what we each were coming up with. We initially wanted our roles to be less specialized, but technical and logistical challenges made this difficult. In particular, the AI tools and interfaces were hard to use without expertise. Max spent a lot of effort trying to get one AI tool (Magenta Studio) to work within Ableton Live, but found it difficult to understand. To make the most of our time and our individual skills, we ended up with a process where Matt generated and curated lyrics (and images for artwork), I worked with AI tools to create audio and midi samples and loops, Brodie recorded vocals, and Max arranged and mixed together all the different "puzzle pieces".

Interestingly, when we came together after the song was finished, we realized that during the process, perhaps because of our different roles, we hadn't all been viewing the AI's role in the collaboration in the same way. Matt saw our process as a back and forth collaboration between the group and the AI: "Much like the call and response style repetitions that occur in the chorus of our song (the human voice being mirrored by the machine-like one), a similar dynamic took place with the AI collaboration. However, in this case it was reversed. The model would output multiple verses and choruses, and we would choose the ones we found most compelling, feeding these back into the model to generate new variations and expansions. Step by step, we found the shape of our song in this back and forth dynamic." Max, however, viewed it another way, saying: "I don't feel like the AI was a collaborator. I feel like the AI was the artist in this one. And maybe it's just because I produce artists that that's the approach I took... The AI said ok, this is my take in the booth, this is the best I can do, I don't really know how to play guitar or anything, but here's my idea... The AI wrote the melody and the lyrics... and then we kind of took the pieces and produced a song, but tried to leave the AI's soul in the song without producing it out." I, on the other hand, felt at the time that we were the real artists, rather than the AI: "From my position, I feel like there's so much of each of our souls in the song. Because if we had given those AI pieces to somebody else, the thing that came out of it would have been completely different."

6.2.2 Creative Process

Choosing a Concept and Artistic Direction

In our first conversations, we discussed the concept for our song. Given our collective uncertainty about what the AI would do, we wanted to establish a clear story and direction that would remain the foundation, regardless of the chaos that was to come. Early on, we settled on building a song that would gradually change, over the course of 4 minutes, from delicate and intimate "music for human ears" into "music for computers to listen to". We decided to begin the first verse with the most organic, intimate human sounds and instruments we could come up with, and then watch what happened as we filtered those sounds through the lens of AI, continually feeding back into itself. We hoped that eventually, much like when the wheel of a car spins so fast that it projects the illusion of turning backwards, something would emerge from this cycle that would be beautiful to our human ears. Though we diverged from this vision, ending up with a more iterative feedback loop between us and AI, many of the choices we made at this stage made it into the final track. The idea for recruiting Brodie to sing came out of the idea to start with an intimate, human arrangement. Max said: *"Brodie has a very emotive, human voice. If I'm thinking of AI, it would be the opposite of Brodie's voice."* The vocoder effect on the vocals plays out this push and pull throughout the track, coming together with Brodie's natural voice at times, and singing call and response at 2:08. We also chose the piano sound at the very start of the song (which comes back at the end at 3:12) for its warmth and intimacy - you can hear the creaking sounds of the piano pedals as well as the breathing and rustling sounds of the pianist on the bench.

In parallel with our vision for the song's sound and structure, we took inspiration from the song "Daisy Bell" (composed by Harry Dacre in 1892), which was the first song to be sung by a computer (the IBM 7094) in 1961. "Daisy Bell" captures some of the sweetness and intimacy that we wanted to go for, and it felt symbolic of the strange juxtaposition between human and machine that we needed to be able to embrace in order to write this song. We went on to use "Daisy Bell" as the initial "seed" for both the lyrics (generated by GPT-2 [173]) and the vocal melodies (generated by Magenta Studio's Continue plugin [75]). In addition, Brodie recorded an A capella version of "Daisy Bell", which I used to train SampleRNN [174] models - these were ultimately used to generate many of the samples used throughout the track.

Assembling AI Puzzle Pieces: Lyrics and Melodies

To generate lyrics, we used a GPT-2 model that had been fine-tuned on a large dataset of song lyrics. We seeded the generation process with the first line from the chorus of "Daisy Bell" ("Daisy, Daisy, give me your answer, do"), and then iteratively selected lyrics we found compelling from the output to re-seed the generation process. Since this model let us condition on year, we used the following years for each successive generation step: 1961 (the year an IBM 7094 first sang these same lyrics), 1984, and 2019 (the most recent year of lyrics that the model was trained on). There were some instances in which the generated lyrics seemed so rich and apt that we had to double check they weren't copied from the original lyrics in the training set (fortunately they weren't). We did have to censor our AI lyricist at times to keep it from veering in directions that made us uncomfortable.

Despite our expectations, the lyric generation turned out to be one of the smoothest and most important AI contributions. As soon as the rest of the group saw the line "Go on, listen to your body choir"(1:04), we knew that was going to be the chorus of the song. Max says: *"Initially I was thinking the AI lyricist would be a very small contribution, and then in the end it ended up being the core of the whole song."* Brodie adds: *"I thought the AI lyrics were fascinating and loved the way they strung together words in ways I would never have imagined. It was challenging to match the AI words to the AI melody - kind of like a brain puzzle trying to match the word syncopations and phrasing to the melody in a way that* *felt right to me. My brain was tired at the end of recording, but it was a fun challenge!"*

As we did with the lyrics, all of the vocal melodies were generated by asking an AI to "continue" the melody from "Daisy Bell", in this case using Magenta Studio's Continue plugin. Because of this choice, our song ended up in the same key as "Daisy Bell", although we changed the tempo and time signature. The verse melodies that come in at 0:01 and at 1:36, as well as the chorus melodies at 1:04 and 2:09 were all created by stitching together sections, usually in pieces lasting about 1 measure, of several AI generated melody takes. Brodie stayed faithful to these generated melodies throughout - the one point when she takes a bit of liberty with the AI melody is in the section at 2:40, as the natural human voice seems to take back control from the AI. We snuck in one other AI-generated melody into the piano line at the very end of the track (3:45). This was created by a Music Transformer [18] model "continuing" Max's piano playing.

More Puzzle Pieces: Beats, Samples, and Feedback Loops Between Humans and AI

While we let the AI lead the way in terms of the lyrics and melody, for the rest of the music creation and production process, we primarily focused on using AI to make sounds, rhythms, and textures that we otherwise would never have created. Our goal was to see if we could lean on AI here for sound design and to create a unique palette that wouldn't sound like anything else. We did this in several forms. First, we trained a version of a GrooVAE [6] model for generating drum loops with a variety of different swing and syncopation feels, interpolating between different loops to produce something that was irregular and surprising. This beat first comes in at 0:32 and again at 1:36. We also turned this beat into the bassline (1:36), which we made by copying the generated kick drum pattern to our bass track, and then moving the MIDI notes to fit with the chord progression without changing their initial timing.

In addition to the beat, which served as one of the first inspirations for the feel of the track along with the AI-generated lyrics and melody, we trained SampleRNN models on a 30-second recording of Brodie singing "Daisy Bell". Because SampleRNN requires much more than 30 seconds of audio to train, we built a dataset of about an hour by making a large variety of pitch-shift and time-stretch adjustments to those 30 seconds. The end result was a model that sounded like it was trying desperately to sing like Brodie, but it was really grasping. We generated, curated, and warped a large number of samples from this SampleRNN to create the palette for the rest of our song. The percussion sounds at 0:32 are all made from these samples, as are the wobbly, vocal-like bits that come in at the same time and persist throughout. The chords that come in at 1:04 are also made from these samples: this model had a tendency to land on one note and sound like it was screaming 'eeee' for about 15 seconds, but in the mix, this pad part ended up actually sounding very smooth and warm. Once we had recorded the vocals actually used in the song and composed more parts around that, we re-trained this model on of our work-in-progress stems (which was easier with more material) to create another round of SampleRNN samples for the rest of the song.

6.2.3 Takeaways and Reflections

Strange, Beautiful, Eerie

After we finished producing the song (but before the contest was judged), I interviewed my three collaborators and wrote reflections on my own experience making the song. A common thread was a feeling of surprise about how strange the process of creating the song was, paired with positive feelings about both the final song itself and the fun process. Brodie said: *"I love how it turned out and how the organic and AI merge together in this song. It's strange and beautiful and a little eerie. I think my favorite part was the wild lyrics that came out of this and how fun they were to sing." Max added: "That was weird. It was a weird project. It was almost like writing a song where my left arm was a cybernetic arm with extra abilities, and my right arm, I wasn't allowed to use."*

Learning to Work with AI and Adapting to Challenges

After investing many hours trying hard to work with the AI tools we had chosen, both Max and I found that using SampleRNN in particular was harder than expected. I found that training models took a long time and that it took a lot of trial and error to find the right seed audio to prime the trained model to do something interesting: *"Even having done this kind of thing before, it turned out to be so hard to get much at all out of the models."* Max was struck by how inhuman the AI model was in the way it perceived audio: *"I think I just underestimated the difference between listenable music for humans and the way a machine interprets sound. I know as an audio engineer, the difference between how you hear someone speak versus when you hear it coming through a microphone - you're like, what's all that extra sound? That's already there but your ear tunes that out. But the microphone just says, I'll take anything, lets get it all! It was interesting seeing all the struggle it took to get anything out of the AI but a screaming 'eeee' sound."*

Despite these struggles, however, all four group members felt like the process eventually led us to a places that we were happy with, ultimately finding unique sounds, melodies, and lyrics that we liked. Brodie felt like using AI prevented her from overanalyzing or second-guessing her lyrical and melodic choices: *"I think the melodies and lyrics the AI came up with were so unique and something I would never have done on my own, so I loved how it kind of broke open my creative box a little. I also tend to spend lots of time agonizing over words or melody, and it was freeing to just go with what the AI created and enjoy adding my organic voice and emotion onto it."* I felt similarly about using drum beats made with AI, writing: *"Normally I'll go down a rabbit hole, thinking I have to move this hi-hat here or there, that it will make so much difference."*

Max felt like using AI forced him to flip the process that he normally uses when making music on the computer: *"If we had written it ourselves, it would have been more loop-based, and we would have been thinking, how do I make this interesting? And this was the opposite direction, where the question is how to make this into a song, because it's too interesting. Because the rhythms were so strange, we didn't have to worry about adding in all these fills or stuff like that. It just kind of kept going, which is pretty cool."* After having been through this process once, he sees it becoming a normal part of his composition process in the future: *"As a producer, sometimes you bring in a guitarist, sometimes you bring in a trumpet player, sometimes you bring in a drummer, and sometimes you bring in an AI.. The AI does this kind of thing.. They're pretty crazy when they go in the vocal booth, but it's pretty cool."*

6.3 Perspectives on AI-Generated Musical Materials: Case Studies with Two Musicians

6.3.1 Introduction

This section explores case studies with two musicians from different backgrounds, capturing their experiences through the process of working with AI to create customized materials for use in new compositions of their own. Building on Sections 6.1 and 6.2, I explored how, by intentionally gathering their own *Collections* of material to use as input to AI, and by working with me serving as an intermediary between them and various AI systems, musicians without a working knowledge of machine learning might still be able to experience meaningful or inspirational creative interactions with AI.

Beyond Prompting: *Collections* as Inputs for Creative Interactions with AI

Generally, users can interact creatively with machine learning models either by **(1)** providing a conditioning input or a "prompt" to a pretrained model, or **(2)** curating one's own training dataset. Developing intuition about what makes a good prompt or a good training dataset, or how to use a model in the first place, however, might be difficult for artists who don't have much experience with AI [175]. This study aims to probe what deeper and more personalized musical interactions with AI might be like without requiring participating musicians to become expert AI users or AI engineers themselves.

With the aim of surfacing early design directions that might better facilitate creative interactions with AI for musicians and artists, I started with a basic concept for a user interaction: the participant (a musician) was asked to send off a personally curated *Collection* of sounds, and they received back customized music composition materials (samples, loops, and digital instruments) put together with the help of AI. These materials were delivered back to the user within a custom template inside a DAW (Digital Audio Workstation), to facilitate easy exploration and to fit within their existing processes. Curating collections of music samples for a project is a common practice already for some musicians or producers; several platforms like Splice and Reason sell "packs" of samples created by a range of different artists and sound designers. Figure 6.2 shows a mockup of this simple starting point for a user interface.

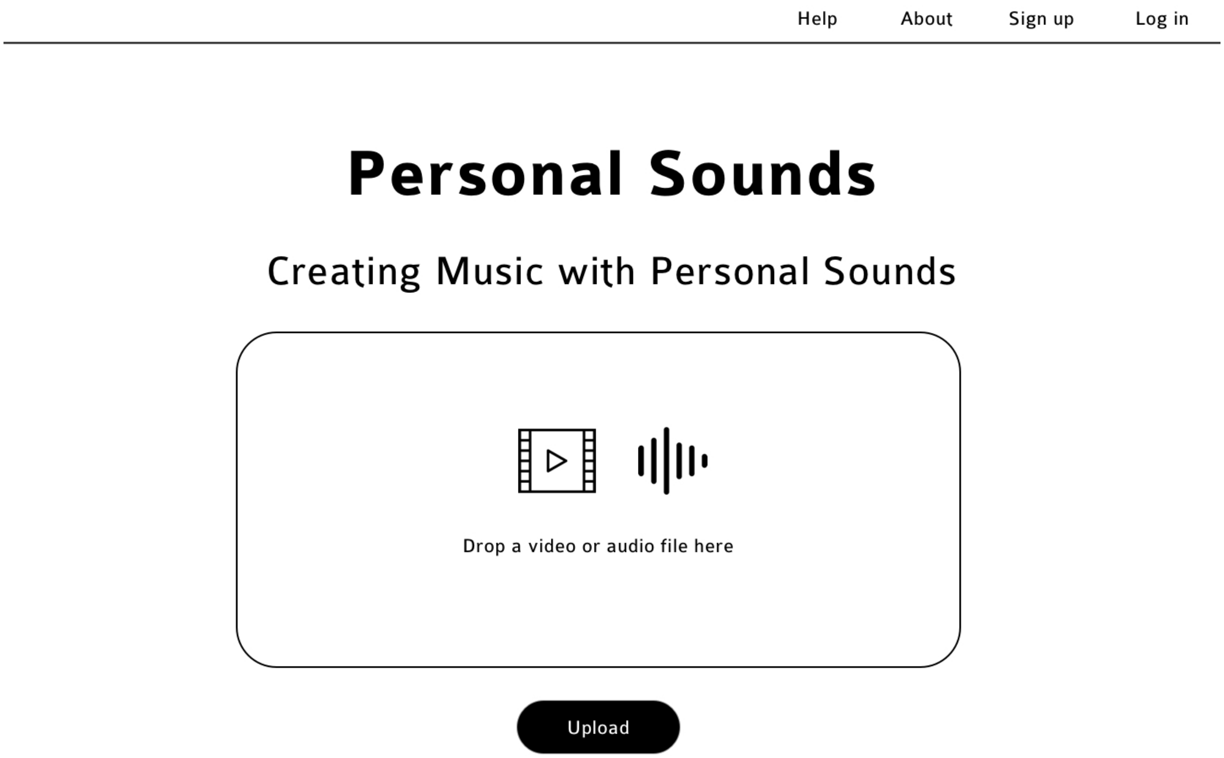

Help
About
Sign up
Log in
Personal Sounds
Creating Music with Personal Sounds
Drop a video or audio file here
Upload

Figure 6.2: Mockup of a web interface demonstrating the look of a platform for creators to upload sounds to be used as inspiration for AI-generated materials.

6.3.2 Study Design

Being the Recording Engineer: Mediating Interactions Between Artists and AI

This study aimed to allow participants to **(1)** focus on curating their *Collection* and **(2)** to give them a chance to play around with AI-generated materials created based on their *Collection*. To keep participants from needing to worry about what the AI itself would be doing, details about what happened between upload time and download time were kept minimal - participants were just told that based on their *Collection*, I would be sending them a customized music template made using AI.

In order to facilitate this study for the participants, I took on the role of managing the engineering tasks involved in working with the currently available range of AI tools for creating music, including data pre-processing, training new models, and running pre-trained models. Similar to the way in which recording or mixing engineers with a working knowledge of studio gear often mediate the experiences of artists when they go into recording studios (so that expertise in audio engineering does not need to be a prerequisite for every musician that wants to record a song), my experience in operating AI technology allowed me to take care of some overhead in order to facilitate interactions between the participants and AI. While AI literacy may eventually become commonplace among artists, today's average artist or musician is unlike to have a working knowledge of how to use it [176].

It is important to mention that in my role as the one responsible for operating the AI technology, I also needed to play an artistic role to some extent, for example by choosing which model to apply to which sounds or by filtering out less promising AI-generated outputs. The recording engineer analogy still applies here - successful audio engineers are often highly valued by artists because of their taste in making decisions about how to apply the gear in the studio, such as choosing a microphone that fits well with a particular singer's voice [177]. Together with the two musicians who participated, I followed a three-step process:

1. **Collecting Sounds:** Each participant spent a period of a week or two looking for sounds that were meaningful to them and that they were interested in using as source material for creating something new during the course of this study. They shared their audio files with me along with short descriptions explaining why they chose each one.

2. **Creating a Customized Music Template with AI Generated Materials:** Using the sounds that they provided to me, I created a set of musical materials for each participant using several machine learning models. The materials consisted of samples, loops, and digital instruments, and were delivered to the participants as a file within their preferred Digital Audio Workstation.

3. **Composing:** Using the materials provided in their custom AI template, each participant created a sketch for a new composition of their own and shared it with me.

After each step of the process, I asked the participants about their experiences and perspectives through semi-structured interviews. The rest of this section reports on the two musicians who worked with me in this way. The names used in this section are pseudonyms.

6.3.3 Case Study 1: Getting Back into Making Music

Ralph's Musical Background and Motivation

Ralph is in his 50's and has several decades of musical experience in various roles as a musician, mix engineer, and arranger. He played bass, drums, or keyboards in a number of bands, and he worked as a sound engineer for more than 10 years, recording albums and working as a front-of-house engineer for over 300 shows. In bands, Ralph often functioned as the arranger, and he developed as a composer through writing music with the bands he was part of. For the last 20 years, however, he has been less directly involved in creating music himself, although he works creating new products for audio companies. Recently, Ralph has become more interested in making music again through collaborating with his kids, and in particular his 11 year old son who is learning to make music using the Groovepad and GarageBand apps for the iPad.

Ralph's Sound Collection

For his *Collection*, Ralph chose sounds based on their emotional content. He sent a total of 14 audio files, along with a sentence describing each of them. Ralph chose a few fragments of pieces of music that were important to him (either songs from other artists that influenced him strongly, or recordings that he had made in the past), and he also included recordings of his parents' and his children's voices, either from voicemails or pulled from videos. *"I collected sounds that had high emotional value for me. It included music I listened to... songs that were extremely important for my musical development... I basically picked song segments, or sounds that were emotionally really important... these were all super important for my life, including obviously my kids."*

Putting Together Ralph's AI Music Materials

I created Ralph a customized template in GarageBand, populated with musical materials based on his *Collection*, asking him to play around with it and to try to sketch out an idea for a new song using some of the elements in there.

For the sounds he chose that were snippets of existing songs, I tried to use AI tools to take inspiration from the more memorable aspects of those snippets. For example, Ralph sent me a few seconds from two different Kraftwerk songs, explaining: *"Even though Kraftwerk was not a pop band, they were extremely good at simple melodies, and this one was the best example. It triggered in me the focus on ear work music composition."* In response, I used a MusicVAE model [26] to generate several "interpolations", asking the model to generate a range of melodies that were somewhere "between" the melodies from Ralph's influences. He also sent me the first few seconds from another song called "Take Me Up", explaining: *"This is the beginning of a song which was critical in my mid-20s, where I went on a beach vacation with friends on the Canary Islands, and projected my feelings very deeply into some of the dominant songs during that time."* To capture something of the timbre of this song, I use a source separation model to extract the synthesizer sound, which I imported into a sampler to be playable as in instrument in GarageBand.

I treated the non-musical files that Ralph sent me as part of his *Collection* (like his children's voices) differently. For these sounds, I wanted to find musicality in them while still respecting Ralph's desire not to transform those personally meaningful sounds into something that he wouldn't recognize. I implemented this approach by finding moments from the recordings of voices that happened to have a consistent enough rhythm to turn into a loop, adding these loops into Ralph's GarageBand file.

Findings

Ralph's experience was shaped largely by how clear the connections were between what he had sent me and the samples and instruments that I sent back to him. Ralph

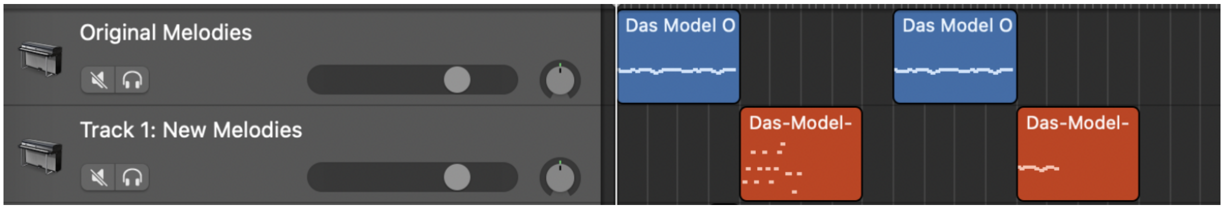


Figure 6.3: Two tracks from Ralph's customized Garageband File

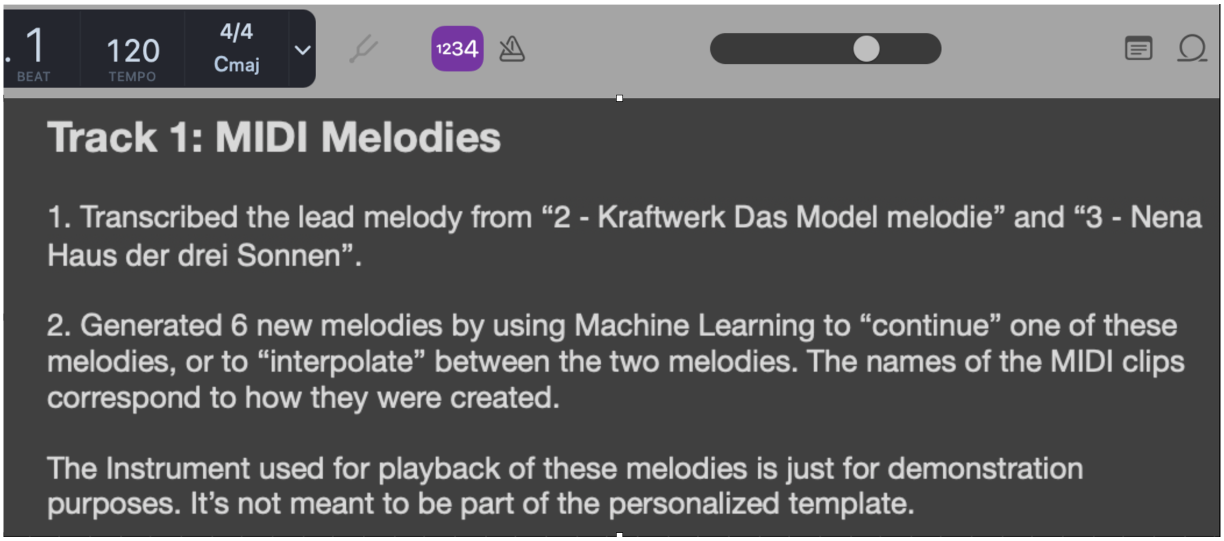
had chosen specific sounds from specific memories, and before working with the materials I sent him, he asked for clarification, wanting to be able to tell more clearly how the sounds and instruments that I sent back to him had been related to the ones he sent me. In response, I reorganized his GarageBand file, adding in his original sounds together with the new AI-generated material to provide context. Figure 6.3 shows two tracks from Ralph's template, with a melody he provided from the Kraftwerk song "Das Model" shown in blue, and new melodies generated based on that melody shown alongside it in red. I used the "Track Notes" feature (which many applications like GarageBand offer) to summarize the series of steps used to create the material in each track. Figure 6.4 shows a screenshot of those notes.

After taking some time to compose a sketch of a new song using his GarageBand template, Ralph felt that the melodies, samples, or other materials that he wanted to use the most were the ones that were most connected to his original audio files. In his piece, he featured the generated melodies based on Kraftwerk and his other influences, a loop containing his kids voices, and he also added a few other instruments and sounds from the library that comes with GarageBand.

He also used some of his original samples without the modifications I made. One sound he included was the sample of a car door closing from the beginning of Kraftwerk's song "Autobahn", which I had turned into a playable drum kit for him by lowering the pitch and adding effects. *"If you transform it, very soon you can't recognize the origin anymore, and I wanted to have the origin in there. So that's why I felt I wanted to have the original sample somewhere. Even if it's just the beginning of the song where you have the originals.. and then you vary it so that it becomes the instrument. I think making this progression clear within the song would have been a goal that I would want to have."*

For Ralph, the idea of transforming the sounds with AI shaped the structure of the song he ended up wanted to make in the end: *"I wanted to create a song where even the extraction process itself becomes clear, otherwise I would have to explain it in text or in interviews why this is so important... ideally I would have explained the origins of these sounds within the song itself versus just knowing this is a really important song to me because it consists of all these sounds."*

While Ralph found some parts of this exercise confusing, he ultimately ended up liking the song that he made, listening to it multiple times and even playing it for his son in the car. Ralph's experience highlights a theme that also came up in Section 6.1 - when working with samples that have stories attached to them, preserving that story was important. Telling a new story through the sounds and instruments created



Track 1: MIDI Melodies

1. Transcribed the lead melody from "2 - Kraftwerk Das Model melodie" and "3 - Nena Haus der drei Sonnen".

2. Generated 6 new melodies by using Machine Learning to "continue" one of these melodies, or to "interpolate" between the two melodies. The names of the MIDI clips correspond to how they were created.

The Instrument used for playback of these melodies is just for demonstration purposes. It's not meant to be part of the personalized template.

Figure 6.4: Notes shown in context within the GarageBand file indicating the sequence of transformations to create the material

by AI was interesting, but for Ralph, he preferred to make it clear where that story's origin was by showing the transformation from his original sample to the AI-generated sample as it happened through the course of his song.

6.3.4 Case Study 2: Going Deeper with Sound Design

Alvaro's Musical Background and Motivation

Alvaro is in his 20s and works as a professional musician, producer, and sound designer. He was interested in participating in this study because he saw the potential of AI for creating new materials in his sound design process. He has professional experience designing sample packs for other music producers, and he had also recently taken a six-week course for musicians learning how to use AI for music and sound.

Alvaro's Sound Collection

Rather than choosing sounds primarily based on memories like Ralph did, Alvaro came up with his *Collection* based on his recent work in sound design and music production. He sent me a set of 10 audio files that were about 30 seconds long on average. He said: *"These are a combination of processed field recordings through Cecilia 5 and stuff I put together with my modular with a bit of post-processing in Ableton. I would say they are a good representation of where I am aesthetically and emotionally in regards to sounds and sonic textures."* Alvaro's sounds included two drum beats as well as several highly processed field recordings from places he had been, including the sound of a crowd and the sound of his washing machine. In Alvaro's description of the sounds he chose, he shared less with me about the places they had been recorded, instead telling me about the timbres and textures. Some of the original sources, like the washing machine, were difficult for me to recognize when he sent them to me, because he had done a lot of postprocessing to turn the sounds into interesting, abstract rhythmic textures.

Putting Together Alvaro's AI Music Materials

Based on his interests as a sound designer, as well as the nature of the sounds he provided, which were rich and detailed in texture, I decided to train a SampleRNN [174] model (a generative model of raw audio) for him in order to create a range of variations of his sounds that would not have been possible to produce using his usual music production tools. Because of Alvaro's interest in experimentation with timbre, he seemed to be looking for new kinds of sounds that he had never heard before. Deep learning models like SampleRNN usually produce imperfections and artifacts that make their outputs sound different from their source material.

In order to get enough training data to reasonably train this model (the implementation that I used suggests using at least 1 hour of audio), I needed to first apply a large amount of data augmentation to his audio samples, which I did by changing the pitch and timing. After generating a large number of new samples for Alvaro, I listened through, choosing what I thought were some of the more unique and surprising audio outputs. I put these samples together for Alvaro in a file within Ableton Live (the music software he uses), and I played the samples for him one by one over a video call as he listened to them for the first time.

Findings

Alvaro was quick to imagine different possibilities of what he might do with his customized materials - e.g. *"That was a pad or chord sound, definitely."* He compared these sounds to what he might normally get in a professional sample pack, highlighting that while sample packs are highly organized and tagged with information like musical tempo and key, these samples were much more unpredictable. He liked that there were *"artifacts"* and *"glitches"* in the sounds that made them unique. *"With this, for me it's cool to just get like a snippet of sound and you can kind of take it to wherever you want. You can just pitch it up or down, accommodate it to whatever pitch or tempo, stretch it."* Listening to the sounds, Alvaro imagined himself and the AI as both being chefs, coming up with ingredients to cook with together.

Alvaro was enthusiastic about the sound samples he received, and he was interested in learning more about how I made them. While Alvaro had actually learned about SampleRNN once before, when he attended a workshop on using AI for music and sound design, he did not find it especially useful or interesting at the time, because it was slow, needing hours to train. At the workshop, he had also not tried curating his own personal *Collection* of audio to train the model. This time, however, when I trained the model for him using his own *Collection*, he found the results produced by the model to be very exciting. *"its almost like sampling an old record, despite it being an AI... "It becomes like an extension of myself, and it comes up with stuff I could have come up* *with but with its own twist, so I feel like its still aesthetically connected, but it's like a remix of myself, so it's like looking at myself in the mirror."*

After receiving his AI music materials, Alvaro composed a sketch for a new track using only the samples I had sent him (generated by SampleRNN) as source material. He spent an hour composing this sketch, finding that it was engaging material to work with and also that after going through the entire process once, he now had new ideas for sounds to use for his next *Collection*.

6.3.5 Conclusions

In this study, I explored possibilities for a collaborative creative process between an artist and an AI Music Engineer with two musicians participating, Ralph and Alvaro. My findings suggest that although access to AI technology holds potential for creators looking for new ways to produce music or design sounds, the usability of currently available tools is a major roadblock that prevents people like Ralph and Alvaro from being able to see much of that potential by themselves. While faster models and clearer user interfaces might allow musicians to get more use out of machine learning, access to training and educational resources on machine learning for artists also appears to be an important piece of the puzzle for artists like Alvaro who have an interest in the technology. For artists who are just getting started with machine learning or who have less interest in developing their own expertise, this study also points to the potential for collaboration with an expert in AI and machine learning (an "AI Music Engineer"), who might be able to help guide the artist through the landscape of models or methods are available to try, and give advice about which might be the right tool for the job in a given situation.

Bibliography

[1] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[2] N. Statt, "Jay z tries to use copyright strikes to remove deepfaked audio of himself from youtube," *The Verge*, 2020.

[3] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4779–4783, IEEE, 2018.

[4] M. L. Gray and S. Suri, *Ghost work: How to stop Silicon Valley from building a new global underclass*. Eamon Dolan Books, 2019.

[5] B. L. Sturm, M. Iglesias, O. Ben-Tal, M. Miron, and E. Gómez, "Artificial intelligence and music: open questions of copyright law and engineering praxis," in *Arts*, vol. 8, p. 115, Multidisciplinary Digital Publishing Institute, 2019.

[6] J. Gillick, A. Roberts, J. Engel, D. Eck, and D. Bamman, "Learning to groove with inverse sequence transformations," in *International Conference on Machine Learning*, pp. 2269–2279, 2019.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[8] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," *arXiv preprint arXiv:1808.06601*, 2018.

[9] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2image: Conditional image generation from visual attributes," in *European Conference on Computer Vision*, pp. 776–791, Springer, 2016.

[10] I. Simon, A. Roberts, C. Raffel, J. Engel, C. Hawthorne, and D. Eck, "Learning a latent space of multitrack measures," *arXiv preprint arXiv:1806.00195*, 2018.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[12] O. Senn, L. Kilchenmann, T. Bechtold, and F. Hoesl, "Groove in drum patterns as a function of both rhythmic properties and listeners' attitudes," *PloS one*, vol. 13, no. 6, p. e0199604, 2018.

[13] M. Wright and E. Berdahl, "Towards machine learning of expressive microtiming in brazilian drumming.," in *ICMC*, Citeseer, 2006.

[14] A. Tidemann, P. Öztürk, and Y. Demiris, "A groovy virtual drumming agent," in *International Workshop on Intelligent Virtual Agents*, pp. 104–117, Springer, 2009.

[15] Y. Gu and C. Raphael, "Modeling piano interpretation using switching kalman filter.," in *ISMIR*, pp. 145–150, 2012.

[16] Y. Gu and C. Raphael, "Creating expressive piano performance using a low-dimensional performance model," in *Proceedings of the 2013 Sound and Music Computing Conference*, vol. 61, p. 62, 2013.

[17] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications*, pp. 1–13, 2018.

[18] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," *arXiv preprint arXiv:1809.04281*, 2018.

[19] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the maestro dataset," *arXiv preprint arXiv:1810.12247*, 2018.

[20] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[21] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," *arXiv preprint arXiv:1808.01410*, 2018.

[22] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018.

[23] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," *arXiv preprint arXiv:1808.10128*, 2018.

[24] A. Tidemann and Y. Demiris, "Groovy neural networks.," in *ECAI*, pp. 271–275, 2008.

[25] I. Simon and S. Oore, "Performance rnn: Generating music with expressive timing and dynamics," *Magenta Blog*, 2017.

[26] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," *arXiv preprint arXiv:1803.05428*, 2018.

[27] M. Wherry, "All about quantise," *Sound on Sound*, 2006.

[28] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications," *arXiv preprint arXiv:1701.05517*, 2017.

[29] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," *International Conference on Learning Representations*, 2018.

[30] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," *arXiv preprint arXiv:1701.06548*, 2017.

[31] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Counterpoint by convolution," *arXiv preprint arXiv:1903.07227*, 2019.

[32] D. Ippolito, A. Huang, C. Hawthorne, and D. Eck, "Infilling piano performances," *Workshop on Creativity and Design, NeurIPS 2018*, 2018.

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: a system for large-scale machine learning.," in *OSDI*, vol. 16, pp. 265–283, 2016.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[36] C. Muchnik, M. Hildesheimer, M. Rubinstein, M. Sadeh, Y. Shegter, and B. Shibolet, "Minimal time interval in auditory temporal resolution.," *The Journal of auditory research*, vol. 25, no. 4, pp. 239–246, 1985.

[37] P. Kumar, H. K. Sanju, and J. Nikhil, "Temporal resolution and active auditory discrimination skill in vocal musicians," *International archives of otorhinolaryngology*, vol. 20, no. 4, pp. 310–314, 2016.

[38] Y.-H. An, S. Y. Jin, S. W. Yoon, and H. J. Shim, "The effects of unilateral tinnitus on auditory temporal resolution: gaps-in-noise performance," *Korean journal of audiology*, vol. 18, no. 3, p. 119, 2014.

[39] N. Trieu and R. M. Keller, "Jazzgan: Improvising with generative adversarial networks," *Workshop on Musical Metacreation*, 2018.

[40] A. Tidemann and Y. Demiris, "Imitating the groove: Making drum machines more human," in *AISB*, 2007.

[41] K. Hellmer and G. Madison, "Quantifying microtiming patterning and variability in drum kit recordings: A method and some data," *Music Perception: An Interdisciplinary Journal*, vol. 33, no. 2, pp. 147–162, 2015.

[42] J. Gillick, C.-E. Cella, and D. Bamman, "Estimating unobserved audio features for target-based orchestration," in *ISMIR*, 2019.

[43] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on signal processing*, vol. 45, no. 2, pp. 434–444, 1997.

[44] S. Araki, F. Nesta, E. Vincent, Z. Koldovskỳ, G. Nolte, A. Ziehe, and A. Benichoux, "The 2011 signal separation evaluation campaign (sisec2011):-audio source separation," in *International Conference on Latent Variable Analysis and Signal Separation*, pp. 414–422, Springer, 2012.

[45] Y. Yaslan and Z. Cataltepe, "Audio music genre classification using different classifiers and feature selection methods," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 2, pp. 573–576, IEEE, 2006.

[46] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," *In Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017.

[47] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 2, pp. II753–II756, IEEE, 2000.

[48] E. Humphrey, S. Durand, and B. McFee, "Openmic-2018: an open dataset for multiple instrument recognition," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.

[49] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," *In Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.

[50] C. Southall, R. Stables, and J. Hockman, "Automatic drum transcription using bi-directional recurrent neural networks.," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pp. 591–597, 2016.

[51] G. Carpentier, G. Assayag, and E. Saint-James, "Solving the musical orchestration problem using multiobjective constrained optimization with a genetic local search approach," *Journal of Heuristics*, vol. 16, no. 5, pp. 681–714, 2010.

[52] J. Reiss and Ø. Brandtsegg, "Applications of cross-adaptive audio effects: automatic mixing, live performance and everything in between," *Frontiers in Digital Humanities*, vol. 5, p. 17, 2018.

[53] B. De Man and J. D. Reiss, "A semantic approach to autonomous mixing," *Journal on the Art of Record Production (JARP)*, 2013.

[54] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," *In Proceedings of the 12th International Society for Music Information Retrieval Conference*, pp. 591—596, 2011.

[55] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 631–648, 2018.

[56] Z.-C. Fan, Y.-L. Lai, and J.-S. R. Jang, "Svsgan: Singing voice separation via generative adversarial network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 726–730, IEEE, 2018.

[57] A. Antoine and E. Miranda, "A perceptually orientated approach for automatic classification of timbre content of orchestral excerpts," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, pp. 3723–3723, 2017.

[58] M. Caetano, A. Zacharakis, I. Barbancho, and L. J. Tardón, "Leveraging diversity in computer-aided musical orchestration with an artificial immune system for multi-modal optimization," *Swarm and Evolutionary Computation*, 2019.

[59] S. McAdams, "Perspectives on the contribution of timbre to musical structure," *Computer Music Journal*, vol. 23, no. 3, pp. 85–102, 1999.

[60] S. McAdams, "Timbre as a structuring force in music," in *Proceedings of Meetings on Acoustics ICA2013*, vol. 19, p. 035050, ASA, 2013.

[61] A. Mehrabi, K. Choi, S. Dixon, and M. Sandler, "Similarity measures for vocal-based drum sample retrieval using deep convolutional auto-encoders," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 356–360, IEEE, 2018.

[62] C.-E. Cella and P. Esling, "Open-source modular toolbox for computer-aided orchestration," in *Timbre conference*, 2018.

[63] G. Carpentier, D. Tardieu, J. Harvey, G. Assayag, and E. Saint-James, "Predicting timbre features of instrument sound combinations: application to automatic orchestration," *Journal of New Music Research*, vol. 39, no. 1, pp. 47–61, 2010.

[64] R. Wöhrmann and G. Ballet, "Design and architecture of distributed sound processing and database systems for web-based computer music applications," *Computer Music Journal*, vol. 23, no. 3, pp. 73–84, 1999.

[65] G. Carpentier, *Approche computationnelle de l'orchestration musciale-Optimisation multicritère sous contraintes de combinaisons instrumentales dans de grandes banques de sons*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2008.

[66] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, pp. 18–25, 2015.

[67] F. Chollet *et al.*, "Keras: The python deep learning library," *Astrophysics source code library*, pp. ascl–1806, 2018.

[68] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[69] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," *arXiv preprint arXiv:1511.06391*, 2015.

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[71] J. Gillick and D. Bamman, "What to play and how to play it: Guiding generative music models with multiple demonstrations," in *NIME 2021*, 4 2021. https://nime.pubpub.org/pub/s3x60926.

[72] Z. Zukowski and C. Carr, "Generating black metal and math rock: Beyond bach, beethoven, and beatles," *arXiv preprint arXiv:1811.06639*, 2018.

[73] B. L. Sturm, O. Ben-Tal, U. Monaghan, N. Collins, D. Herremans, E. Chew, G. Hadjeres, E. Deruty, and F. Pachet, "Machine learning research that matters for music creation: A case study," *Journal of New Music Research*, vol. 48, p. 36–55, Mar 2018.

[74] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, and C. Cai, "AI song contest: Human-AI co-creation in songwriting," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, (Montreal, Canada), pp. 708–716, ISMIR, Oct. 2020.

[75] A. Roberts, J. Engel, Y. Mann, J. Gillick, C. Kayacik, S. Nørly, M. Dinculescu, C. Radebaugh, C. Hawthorne, and D. Eck, "Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live," in *Proceedings of the 6th International Workshop on Musical Metacreation*, (Charlotte, United States), p. 7, MUME, June 2019.

[76] F. Pachet, "The continuator: Musical interaction with style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, 2003.

[77] B. Sturm, J. F. Santos, O. Ben-Tal, and I. I. Korshunova, "Music Transcription Modelling and Composition Using Deep Learning," in *Proceedings of the 1st Conference on Computer Simulation of Musical Creativity*, (Huddersfield, UK), p. 6, CSMC, Oct. 2016.

[78] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep Music Analogy Via Latent Representation Disentanglement," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, (Delft, The Netherlands), pp. 596–603, ISMIR, Nov. 2019.

[79] R. Yang, T. Chen, Y. Zhang, and gus xia, "Inspecting and Interacting with Meaningful Music Representations using VAE," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 307–312, June 2019.

[80] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2020.

[81] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, (Montreal, Canada), pp. 662–669, ISMIR, Oct. 2020.

[82] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, G. Xia, and J. Zhao, "PianoTree VAE: Structured representation learning for polyphonic music," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, (Montreal, Canada), pp. 368–375, ISMIR, Oct. 2020.

[83] D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam, "VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, (Delft, The Netherlands), pp. 908–915, ISMIR, Nov. 2019.

[84] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Score and performance features for rendering expressive music performances," in *Proceedings of the Music Encoding Conference*, 2019.

[85] Apple, "In the studio with grammy-nominated music producer oak felder," *YouTube*, 2020.

[86] N. Tokui, "Towards democratizing music production with ai-design of variational autoencoder-based rhythm generator as a daw plugin," *arXiv preprint arXiv:2004.01525*, 2020.

[87] G. Vigliensoni, L. McCallum, and R. Fiebrink, "Creating latent spaces for modern music genre rhythms using minimal training data," *In Proceedings of the 11th International Conference on Computational Creativity*, 2020.

[88] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[89] C. Benetatos, J. VanderStel, and Z. Duan, "Bachduet: A deep learning system for human-machine counterpoint improvisation," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2020.

[90] C. Donahue, I. Simon, and S. Dieleman, "Piano genie," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pp. 160–164, 2019.

[91] K. Chen, C. i Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music SketchNet: Controllable music generation via factorized representations of pitch and rhythm," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, (Montreal, Canada), pp. 77–84, ISMIR, Oct. 2020.

[92] D. J. Levitin, "Control parameters for musical instruments: a foundation for new mappings of gesture to sound," *Organised Sound*, vol. 7, no. 2, pp. 171–189, 2002.

[93] R. Fiebrink, D. Trueman, and P. R. Cook, "A Meta-Instrument for Interactive, On-the-Fly Machine Learning," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 280–285, June 2009.

[94] J. Françoise, "Gesture–sound mapping by demonstration in interactive music systems," in *Proceedings of the 21st ACM international conference on Multimedia*, pp. 1051–1054, 2013.

[95] O. Fried and R. Fiebrink, "Cross-modal sound mapping using deep learning," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 531–534, June 2013.

[96] C. A. Huang, D. Duvenaud, K. C. Arnold, B. Partridge, J. W. Oberholtzer, and K. Z. Gajos, "Active learning of intuitive control knobs for synthesizers using gaussian processes," in *Proceedings of the 19th international conference on Intelligent User Interfaces*, pp. 115–124, ACM, 2014.

[97] M. Dinculescu, J. Engel, and A. Roberts, eds., *MidiMe: Personalizing a MusicVAE model with user data*, 2019.

[98] T. Lucas and J. Verbeek, "Auxiliary guided autoregressive variational autoencoders," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 443–458, Springer, 2018.

[99] J. Gillick, J. Yang, C.-E. Cella, and D. Bamman, "Drumroll please: Modeling multi-scale rhythmic gestures with flexible grids," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, 2021.

[100] M. Fisher, "Burial: Unedited transcript," *The Wire*, Dec 2012.

[101] A. V. Frane, “Swing rhythm in classic drum breaks from hip-hop’s breakbeat canon,” *Music Perception: An Interdisciplinary Journal*, vol. 34, no. 3, pp. 291–302, 2017.

[102] A. Roberts, J. Engel, Y. Mann, J. Gillick, C. Kayacik, S. Nørly, M. Dinculescu, C. Radebaugh, C. Hawthorne, and D. Eck, “Magenta Studio: Augmenting creativity with deep learning in Ableton Live,” in *Proceedings of the International Workshop on Musical Metacreation (MUME)*, 2019.

[103] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, “Computational models of expressive music performance: A comprehensive and critical review,” *Frontiers in Digital Humanities*, vol. 5, p. 25, 2018.

[104] Z. Shi, C. Cancino-Chacón, and G. Widmer, “User curated shaping of expressive performances,” *arXiv preprint arXiv:1906.06428*, 2019.

[105] M. K. Lee, D. Kusbit, E. Metsky, and L. Dabbish, “Working with machines: The impact of algorithmic and data-driven management on human workers,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1603–1612, 2015.

[106] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, and C. J. Cai, “AI Song Contest: Human-AI co-creation in songwriting,” *arXiv preprint arXiv:2010.05388*, 2020.

[107] 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*, MM ’20, (New York, NY, USA), p. 1180–1188, Association for Computing Machinery, 2020.

[108] A. Danielsen, K. Nymoen, E. Anderson, G. S. Câmara, M. T. Langerød, M. R. Thompson, and J. London, “Where is the beat in that note? Effects of attack, duration, and frequency on the perceived timing of musical and quasi-musical sounds,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 45, no. 3, p. 402, 2019.

[109] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, “MidiNet: A convolutional generative adversarial network for symbolic-domain music generation,” *arXiv preprint arXiv:1703.10847*, 2017.

[110] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, “Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer,” *arXiv preprint arXiv:1809.07600*, 2018.

[111] A. Pati, A. Lerch, and G. Hadjeres, “Learning to traverse latent spaces for musical score inpainting,” *arXiv preprint arXiv:1907.01164*, 2019.

[112] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Graph neural network for music score data and modeling expressive piano performance," in *International Conference on Machine Learning*, pp. 3060–3070, PMLR, 2019.

[113] C. Ames, "The Markov process as a compositional model: A survey and tutorial," *Leonardo*, pp. 175–187, 1989.

[114] J. Gillick, K. Tang, and R. M. Keller, "Machine learning of jazz grammars," *Computer Music Journal*, vol. 34, no. 3, pp. 56–66, 2010.

[115] 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, 1994.

[116] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks," in *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, pp. 747–756, 2002.

[117] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, "Music transcription modelling and composition using deep learning," *arXiv preprint arXiv:1604.08723*, 2016.

[118] F. Lerdahl and R. S. Jackendoff, *A Generative Theory of Tonal Music: Reissue, with a New Preface*. MIT Press, 1996.

[119] S. Dixon, W. Goebl, and E. Cambouropoulos, "Perceptual smoothness of tempo in expressively performed music," *Music Perception*, vol. 23, no. 3, pp. 195–214, 2006.

[120] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[121] E. Fu, "Pharrell samples viral clips of police brutality protestors to provide them with song royalties," 2018. Retrieved on Feb 5, 2021.

[122] J. Fallon, "Finneas reveals everyday sounds hidden in "bury a friend" and "bad guy"," *The Tonight Show*, 2020.

[123] M. Lavengood, "What makes it sound'80s? the yamaha dx7 electric piano sound," *Journal of Popular Music Studies*, vol. 31, no. 3, pp. 73–94, 2019.

[124] Amper, "AI Music Composition Tools for Content Creators," *Amper Music*, 2021.

[125] J. Kastrenakes, "TikTok owner may have bought Jukedeck, an AI music startup," *The Verge*, July 2019.

[126] D. Maulsby, S. Greenberg, and R. Mander, “Prototyping an intelligent agent through wizard of oz,” in *Proceedings of the INTERACT’93 and CHI’93 conference on Human factors in computing systems*, pp. 277–284, 1993.

[127] S. Ghose and J. J. Prevost, “Autofoley: Artificial synthesis of synchronized sound tracks for silent videos with deep learning,” *IEEE Transactions on Multimedia*, 2020.

[128] A. Elmsley, R. Groves, and V. Velardo, “Deep adaptation: How generative music affects engagement and immersion in interactive experiences,” in *DMRN+ 12: Digital Music Research Network One-day Workshop 2017*, 2017.

[129] R. Fiebrink and B. Caramiaux, “The machine learning algorithm as creative musical tool,” *arXiv preprint arXiv:1611.00379*, 2016.

[130] R. Louie, A. Coenen, C. Z. Huang, M. Terry, and C. J. Cai, “Novice-ai music co-creation via ai-steering tools for deep generative models,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020.

[131] S. H. Hakimi, N. Bhonker, and R. El-Yaniv, “Bebopnet: Deep neural models for personalized jazz improvisations,” in *ISMIR*, ISMIR, 2020.

[132] C. Elsden, A. C. Durrant, D. Chatting, and D. S. Kirk, “Designing documentary informatics,” in *Proceedings of the 2017 Conference on Designing Interactive Systems*, pp. 649–661, 2017.

[133] J. White, W. Odom, and N. Brand, “Exploring location histories as a design material for reflection with memory compass & memory tracer,” in *Companion Publication of the 2020 ACM Designing Interactive Systems Conference*, pp. 221–226, 2020.

[134] W. Odom, R. Wakkary, J. Hol, B. Naus, P. Verburg, T. Amram, and A. Y. S. Chen, “Investigating slowness as a frame to design longer-term experiences with personal data: A field study of olly,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2019.

[135] A. Y. S. Chen, W. Odom, C. Zhong, H. Lin, and T. Amram, “Chronoscope: A near-eye tangible device for interacting with photos in and across time,” in *Companion Publication of the 2019 on Designing Interactive Systems Conference 2019 Companion*, pp. 1–4, 2019.

[136] K. Ryokai, E. Durán López, N. Howell, J. Gillick, and D. Bamman, “Capturing, representing, and interacting with laughter,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI ’18, (New York, NY, USA), p. 1–12, Association for Computing Machinery, 2018.

[137] K. Ryokai, E. Duran, D. Bseiso, N. Howell, and J. W. Jun, “Celebrating laughter: Capturing and sharing tangible representations of laughter,” in *Proceedings*

of the 2017 ACM Conference Companion Publication on Designing Interactive Systems, pp. 202–206, 2017.

[138] K. Ryokai, J. Park, and W. Deng, "Personal laughter archives: reflection through visualization and interaction," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pp. 115–118, 2020.

[139] W. Odom, M. Yoo, H. Lin, T. Duel, T. Amram, and A. Y. S. Chen, "Exploring the Reflective Potentialities of Personal Data with Different Temporal Modalities: A Field Study of Olo Radio," in *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pp. 283–295, New York, NY, USA: Association for Computing Machinery, July 2020.

[140] W. Odom, R. Wakkary, J. Hol, B. Naus, P. Verburg, T. Amram, and A. Y. S. Chen, "Investigating Slowness as a Frame to Design Longer-Term Experiences with Personal Data: A Field Study of Olly," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), pp. 1–16, Association for Computing Machinery, May 2019.

[141] J. Tibau, M. Stewart, S. Harrison, and D. Tatar, "FamilySong: Designing to Enable Music for Connection and Culture in Internationally Distributed Families," in *Proceedings of the 2019 on Designing Interactive Systems Conference*, DIS '19, (New York, NY, USA), pp. 785–798, Association for Computing Machinery, June 2019.

[142] A. Ståhl, K. Höök, M. Svensson, A. S. Taylor, and M. Combetto, "Experiencing the Affective Diary," *Personal Ubiquitous Comput.*, vol. 13, pp. 365–378, June 2009.

[143] M. Lindström, A. St\a ahl, K. Höök, P. Sundström, J. Laaksolathi, M. Combetto, A. Taylor, and R. Bresin, "Affective Diary: Designing for Bodily Expressiveness and Self-reflection," in *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, (New York, NY, USA), pp. 1037–1042, ACM, 2006.

[144] D. McDuff, A. Karlson, A. Kapoor, A. Roseway, and M. Czerwinski, "AffectAura: An Intelligent System for Emotional Memory," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 849–858, ACM, 2012.

[145] N. Howell, G. Niemeyer, and K. Ryokai, "Life-affirming biosensing in public: Sounding heartbeats on a red bench," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2019.

[146] Amazon, "Amazon, inc. echo," 2020. Retrieved on May 4, 2020.

[147] Google, "Google, inc. nest audio," 2021. Retrieved on Feb 5, 2021.

[148] M. Weiser, "Ubiquitous computing," in *ACM Conference on Computer Science*, vol. 418, pp. 197530–197680, 1994.

[149] W. T. Odom, A. J. Sellen, R. Banks, D. S. Kirk, T. Regan, M. Selby, J. L. Forlizzi, and J. Zimmerman, "Designing for slowness, anticipation and re-visitation: a long term field study of the photobox," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1961–1970, 2014.

[150] P. Sengers, K. Boehner, S. David, and J. Kaye, "Reflective design," in *Proceedings of the 4th decennial conference on Critical computing: between sense and sensibility*, pp. 49–58, 2005.

[151] V. Lai and C. Tan, "On human predictions with explanations and predictions of machine learning models: A case study on deception detection," in *Proceedings of FAT**, 2019.

[152] B. Lubars and C. Tan, "Ask not what ai can do, but what ai should do: Towards a framework of task delegability," in *Proceedings of NeurIPS*, 2019.

[153] Q. Yang, J. Suh, N.-C. Chen, and G. Ramos, "Grounding interactive machine learning tool design in how non-experts actually build models," in *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, (New York, NY, USA), p. 573–584, Association for Computing Machinery, 2018.

[154] P. Khadpe, R. Krishna, L. Fei-Fei, J. Hancock, and M. Bernstein, "Conceptual metaphors impact perceptions of human-ai collaboration," *arXiv preprint arXiv:2008.02311*, 2020.

[155] E. Y. Wu, E. Pedersen, and N. Salehi, "Agent, gatekeeper, drug dealer: How content creators craft algorithmic personas," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, Nov. 2019.

[156] C. J. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G. S. Corrado, M. C. Stumpe, *et al.*, "Human-centered tools for coping with imperfect algorithms during medical decision-making," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2019.

[157] Q. Yang, A. Steinfeld, and J. Zimmerman, "Unremarkable ai: Fitting intelligent decision support into critical, clinical decision-making processes," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2019.

[158] B. Kim, W. M., J. Gilmer, C. C., W. J., , F. Viegas, and R. Sayres, " Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV) ," *ICML*, 2018.

[159] B. Yerak, "Ai helps auto-loan company handle industry's trickiest turn," *The Wall Street Journal*, Jan. 2019.

[160] D. Haraway, "Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspectives," *Feminist Studies*, pp. 575–599, 1988.

[161] S. Bardzell, "Feminist HCI: Taking Stock and Outlining an Agenda for Design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, (New York, NY, USA), pp. 1301–1310, ACM, 2010.

[162] C. D'Ignazio and L. F. Klein, *Data feminism*. The MIT Press, 2020. OCLC: 1157171756.

[163] K. Charmaz, *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*. Pine Forge Press, 2006.

[164] P. Dourish, C. Lawrence, T. W. Leong, and G. Wadley, "On Being Iterated: The Affective Demands of Design Participation," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, (New York, NY, USA), pp. 1–11, Association for Computing Machinery, Apr. 2020.

[165] T. Hirsch, "Practicing Without a License: Design Research as Psychotherapy," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, (Honolulu HI USA), pp. 1–11, ACM, Apr. 2020.

[166] C. Lustig, K. Pine, B. Nardi, L. Irani, M. K. Lee, D. Nafus, and C. Sandvig, "Algorithmic Authority: the Ethics, Politics, and Economics of Algorithms that Interpret, Decide, and Manage," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, (San Jose California USA), pp. 1057–1062, ACM, May 2016.

[167] N. Howell, L. Devendorf, T. A. Vega Gálvez, R. Tian, and K. Ryokai, "Tensions of Data-Driven Reflection: A Case Study of Real-Time Emotional Biosensing," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, (New York, NY, USA), pp. 1–13, Association for Computing Machinery, Apr. 2018.

[168] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, "Model cards for model reporting," in *Proceedings of the conference on fairness, accountability, and transparency*, pp. 220–229, 2019.

[169] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumé III, and K. Crawford, "Datasheets for datasets," *arXiv preprint arXiv:1803.09010*, 2018.

[170] P. D. Miller, *Sound unbound: sampling digital music and culture*. Mit Press, 2008.

[171] M. Youngblood, "Cultural transmission modes of music sampling traditions remain stable despite delocalization in the digital age," *PloS one*, vol. 14, no. 2, p. e0211860, 2019.

[172] M. Schuster, D. Mitchell, and K. Brown, "Sampling increases music sales: An empirical copyright study," *American Business Law Journal*, vol. 56, no. 1, pp. 177–229, 2019.

[173] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *Open AI Blog*, 2019.

[174] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio, "Samplernn: An unconditional end-to-end neural audio generation model," *CoRR*, vol. abs/1612.07837, 2016.

[175] E. Cetinic and J. She, "Understanding and creating art with ai: Review and outlook," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 18, no. 2, pp. 1–22, 2022.

[176] R. Fiebrink, "Machine learning education for artists, musicians, and other creative practitioners," *ACM Transactions on Computing Education (TOCE)*, vol. 19, no. 4, pp. 1–32, 2019.

[177] D. Beer, "The precarious double life of the recording engineer," *Journal for Cultural Research*, vol. 18, no. 3, pp. 189–202, 2014.