

# Living, Singing AI

An evolving, intelligent, scalable, bespoke composition system

*by*

**Manaswi Mishra**

Submitted to the Program in Media Arts and Sciences, School of Architecture  
and Planning in partial fulfilment of the requirements for the degree of

**Master of Science**

at the

**Massachusetts Institute of Technology**

September 2021

© Massachusetts Institute of Technology, 2021. All rights reserved.



Author.....

Program in Media Arts and Sciences  
August 25th, 2021



Certified by.....

Tod Machover  
Muriel R. Cooper Professor of Music and Media  
Thesis Supervisor



Accepted by.....

Tod Machover  
Academic Head, Program in Media Arts and Sciences

# Living, Singing AI

An evolving, intelligent, scalable, bespoke composition system

*by*

Manaswi Mishra

Submitted to the Program in Media Arts and Sciences, School of Architecture  
and Planning in partial fulfilment of the requirements for the degree of

Master of Science

## **Abstract:**

As mathematical ideas from artificial intelligence become more accessible to implement on personal computers and cloud based services, they enter the arsenal of tools for qualitatively innovative forms of human expression. In the creative field of music composition, the initial use of artificial intelligence for imitation will soon give way to new paradigms of companionship where the artificial intelligence systems inspire and assist in the human creative process. This new era of artificial intelligence-driven musical instruments is an opportunity for musicians at all levels of skill and technological prowess across cultures. Accessibility of these early tools however has so far been limited to programmers and very narrow musical domains with minimal control over personalizing the outputs produced.

This thesis proposes *Living, Singing AI* : an accessible and bespoke artificial intelligence composition system that bridges the gap between non-programming musicians and modern generative algorithms. It provides a framework to iteratively generate and shape sonic material using one's voice as an input with the following affordances - a simple non-programming interface, an egalitarian framework of multiple large-scale music synthesis models for brainstorming, and a focus on each individual's sonic ideas and aesthetic preferences to create a personal, intelligent composition model. Using this tool, the composer is able to use their *voice* to generate a wide range of novel sonic construction material using open source music synthesis models with intentionality, and to shape a bespoke intelligent composition system. Through peer workshops and application scenarios in practice we show the potential of democratizing artificial intelligent music tools as a brainstorming system to provoke and direct serendipity - an essential ingredient of creating music.

Thesis advisor:

Tod Machover

Muriel R. Cooper Professor of Music and Media

# Living, Singing AI

An evolving, intelligent, scalable, bespoke composition system

*by*

Manaswi Mishra

This thesis has been reviewed and approved by the following committee member



Tod Machover .....

Muriel R. Cooper Professor of Music and Media

MIT Media Lab

# Living, Singing AI

An evolving, intelligent, scalable, bespoke composition system

*by*

Manaswi Mishra

This thesis has been reviewed and approved by the following committee member

Pattie Maes .....

A handwritten signature in black ink, appearing to be 'Pattie Maes', written over a horizontal dotted line.

Professor of Media Technology  
MIT Media Lab

# Living, Singing AI

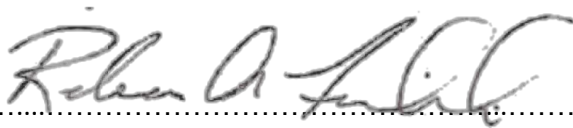
An evolving, intelligent, scalable, bespoke composition system

*by*

Manaswi Mishra

This thesis has been reviewed and approved by the following committee member

Rebecca Fiebrink .....



Reader,

Creative Computing Institute at University of the Arts London

# Acknowledgements

I want to express my deepest gratitude for:

My advisor Tod Machover: Thank you for always inspiring the best in me with your dedicated mentorship, and abundance of care - in person and over zoom. I am ever grateful for your guidance and the opportunities you have given me to participate in groundbreaking projects to leverage my skills and expand my horizons, to grow. Thank you for your trust and belief. My readers Pattie Maes and Rebecca Fiebrink for their deep technical expertise and thoughtful conversations that helped give an early shape to this thesis. Thank you for your time and wisdom.

My Opera of the Future family: Rébecca, Charles, Ben, Nicole, Alexandra, and Nikhil for always inspiring with their immense talent, creativity, and care. Thank you for the many conversations, technical help, and advice on navigating grad school. I hope to continue to learn from and work with each of you in the future. Aarón, Karsten and Hannah for the megadesk camaraderie through a challenging two years – in person: across multiple continents and in virtual: across multiple zoom/telegram spaces. I felt happy and in company with you at my megadesk. Sizi and Priscilla for nurturing and facilitating multiple projects. Thank you for having my back.

My Pandemic family: Vidhi, Dhvani and Nishant for making the pandemic lockdown year in Houston feel like our home was full of life and joy. My friends: Siddharth, Kushagra, Albin and Vibhor for the many conversations and musical jams across continents that always kept sparking my flame.

My Family: Papa and Mummy in Mumbai, and Papa and Mummy in Kota for always being my loudest cheerleaders and believing in my abilities and ambitions. Thank you for all your love and unwavering support.

Sumedha, for your boundless love, kindness, empathy, and strength through all of the furies of creation, the obstacles and the celebrations of accomplishments. I couldn't have climbed a single mountain or written a single song without you by my side and this thesis – just like life, is no different. Thank you.

# Figures

1.1	Overview of the <i>Living, Singing AI</i> system	3
2.1	Score from George Brecht’s process work – ‘Drip Music’ (1962)	5
2.2	(a) Daphne Oram’s glass waveform slides (1957)	6
	(b) Xenakis’ graphic score for Pithoprakta (bars 52-60) (1957)	
2.3	Musikalisches Würfelspiel (1757) 1 <sup>st</sup> edition proposed	7
2.4	List of applications with jigsaw-style composition	8
2.5	List of browser-based composition applications	10
2.6	Performance of Tidal Cycles at an Algorave – Alex McLean	10
2.7	Biophilia 2011 - Björk’s album accompanied by a suite of mobile apps	11
2.8	Google’s Blob Opera	12
2.9	Lo-Fi Player from Magenta	12
2.10	AI Artist Mario Klingemann’s describing latent space exploration as map making	13
2.11	Human curated text prompts: Abraham-AI a visual arts generating DAO	14
2.12	Harshit Agarwal’s <i>The Anatomy Lesson of Dr. Algorithm</i> (2018)	15
	Memo Akten’s <i>Learning to See: Gloomy Day</i> (2017)	
2.13	AI Music industry timeline	17
2.14	Open source AI Music Software timeline	22
2.15	AI-human partnership paradigms in AI Song Contest (2020)	23
2.16	24 emotions mapped from short Vocalized queries (QBV)	27
3.1	Listen to the Listener: Attentive Generative music system	29
3.2	Design Experiment: Blobs controlled by voice	31
4.1	Overview of the <i>Living, Singing AI</i> system	37
4.2	Visual representation of an example ‘ <i>Organism</i> ’	39
4.3	Web application client - server model	39
4.4	Client side system flow diagram	40
4.5	Server side system flow diagram	41
4.6	User interface view	43
4.7	Blob shape representations of behavior	44
4.8	OpenAI Jukebox hierarchical VQ-VAE structure	46
4.9	Google Magenta’s Differential Digital Signal Processing architecture	47
4.10	Generation Schematic from Seed to Sprouts	49
4.11	Schematic Diagram of Creating your Organism by input Seed idea and Grafting controls	50
4.12	A selected Sprout and its accompanying Grafting Controls (C1 and C2)	51
4.13	Schematic Diagram of Performing with your Organism with Behavior Controls	53

4.14	User Interface view showing Behavior Controls in Performance mode	54
4.15	Chromagrams for two different Sprout ideas corresponding to the same Seed idea	55
4.16	System diagram of performance sub-system	56
5.1	Quantitative Evaluation schematic	60
5.2	Visual representation of dimension reduction through t-SNE	61
5.3	t-SNE visualization of Jukebox samples from the training set across genres	63
5.4	t-SNE visualizations of the Jukebox possibility space	64
5.5	Spectrogram representations of Original Seed idea and Jukebox generated Sprout idea	65
5.6	Spectrogram representations of Original singing Seed audio generated from DDSP	67
5.7	Music Transformer generated Sprouts in their MIDI representations	68
5.8	Visual representation of the AI Software possibility space	71



# Contents

1	Introduction .....	1
1.1	Objectives .....	3
1.2	Thesis Structure .....	4
2	Background and Precedents.....	5
2.1	Algorithms in Music Composition.....	5
2.1.1	Generative Composition Tools .....	7
2.1.2	Interactive Compositions .....	11
2.2	Creative Machine Learning .....	13
2.2.1	Latent Space Explorer <i>Artist</i> .....	13
2.2.2	Prompt Creator <i>Artist</i> . .....	13
2.2.3	Curating Dataset <i>Artist</i> .....	14
2.2.4	Accessibility of AI Creative Tools .....	15
2.3	Artificial Intelligence in Music: Overview .....	16
2.3.1	Timeline of AI Music Entities .....	16
2.3.2	Timeline of Generative AI Music Software.....	19
	A. Early methods.....	19
	B. Data driven methods .....	21
	C. Datasets .....	21
	D. Popular Open-Sourced Models.....	22
2.4	AI Song Contest .....	23
2.4.1	Common Strategies for the AI Song Contest .....	24
2.4.2	Limitations .....	24
2.5	Voice as Input.....	26
2.5.1	Query by Humming.....	27
3	Initial Explorations.....	28
3.1	Flexible Inputs Experiment.....	28

3.2	Slow Creator Experiment.....	30
3.3	Visual Design Experiment.....	31
3.4	Learnings from Experiments.....	32
4	<b>Living, Singing AI.....</b>	<b>33</b>
4.1	Living Singing AI – Philosophy.....	33
4.1.1	Primary Function.....	34
4.1.2	Design Choices .....	34
4.1.3	Dimensions of Novelty.....	35
4.2	Definitions.....	36
4.3	System Overview.....	38
4.3.1	Web Application.....	39
	A. Client.....	40
	B. Server.....	41
4.3.2	Data Model.....	41
4.3.3	User Interface.....	42
4.4	Underlying Models.....	45
4.4.1	AI Models for Generation.....	45
4.4.2	Generation Method.....	48
4.5	Creating <i>your</i> Organism .....	49
4.5.1	Vocalized Control.....	50
4.5.2	Grafting Control.....	51
4.6	Performing <i>with</i> your Organism.....	52
4.6.1	Objective.....	52
4.6.2	Behavior Control.....	53
4.6.3	Performance Sub-System.....	54
4.7	Proliferation of your Organism.....	56
4.7.1	Iteration using Audio Similarity.....	57
4.8	Key Contributions.....	58
5	<b>Evaluation and Discussion.....</b>	<b>59</b>
5.1	Evaluation Model.....	59
5.1.1	Quantitative.....	60
5.1.2	Qualitative.....	61
5.2	Possibility Spaces.....	62
5.2.1	Sonic Possibility Spaces.....	62
5.2.2	AI Software Possibility Spaces.....	69

5.3	Sound Collection Experiments .....	72
5.3.1	Musical vs Non-Musical.....	72
5.3.2	Non-Western Music.....	73
5.3.3	Non-Verbal Utterances .....	74
5.3.4	Seed to Composition.....	75
5.4.5	Summary.....	75
6	Conclusion.....	77
6.1	Contributions.....	77
6.2	Future Work.....	78
Appendix A:	Organism Audio Examples.....	81
Appendix B:	List of AI Music Softwares.....	85
References.....		89

# Chapter 1

## Introduction

*“There is an idea, the basis of an internal structure, expanded and split into different shapes or groups of sound constantly changing in shape, direction, and speed, attracted and repulsed by various forces. The form of the work is a consequence of this interaction”*

- Edgard Varèse, (1939)

The word ‘Algorithm’ is derived from the concepts of algebra and arithmetic from mathematicians as early as the 9th century AD from Arabic and Greek to Indian cultures [1]. Algorithmic Music has a similarly long history in the pre-computing era from the ‘process’ works of George Brecht (Drip Music, 1962) [2], Stockhausen (Setz die Segel zur Sonn, 1970) [3], Xenakis (1971) [4] to the US League of Automatic Composers (1978) [5], George Lewis (Voyager) [6], etc.

Artificial Intelligence (AI) algorithms are one of the most widely used and powerful technologies of the twenty-first century so far. Though the exact definition of AI has constantly evolved [7] from the early 1950s of Alan Turing to current *intelligent* Deep Learning architectures like Alpha Go [8], an increase in accessibility has led to a faster assimilation of a wide variety of AI algorithms into our culture. In the creative fields, AI is being used to generate movie scripts (Ross Goodwin’s Spotlight [9]), cooking recipes (IBM’s Chef Watson), poetry (<http://botpoet.com/>), visual art (Portrait of Edmond Belamy, that sold for \$432,500 [10]) etc. There is also a wave of mass generation of AI artwork using open access datasets and open source codebases like the Generative Adversarial Networks (GANs) and their variations [11].

Such AI models for creative endeavors are currently in their infancy and almost exclusively follow the pattern of imitating the dataset they are trained on. In the field of music-making, such models of automated music generation are targeted towards mass generation of music content. This has led to a lot of examples of generative music systems that produce music as a well defined optimization problem. Bach chorales (Deep Bach [12]), Folk Music Modeling (Folk RNN [13]) etc. are some examples of such highly constrained models that acquire representations of patterns that they are trained on.

Modeling music is a hard problem because of two main reasons:

- (a) Semantic hierarchies are complex, subjective and work over multiple scales of time (in the order of seconds to several minutes) [14];

(b) Audio representations at the sampling rates of human hearing make the sequences to be modelled really long, i.e. a typical 4-minute song at CD quality (44 kHz, 16-bit) has over 10 million timesteps.

A lot of progress has been made in the last five years in modeling music for generative systems in its symbolic and waveform representations. Google's Magenta (2016 - 2020), OpenAI's MuseNet (2019) and Jukebox (2020), Sony CSL's Flow Machines (2012), Facebook AI Research's Universal Music Translator (2019) and Amazon's Deep Composer (2019) are some of the efforts from big technology companies utilizing massive resources such as large servers and big datasets. This year (2021) has also seen the growth of academic discourse of AI in computational creativity contexts with editions of Machine Learning for Creativity workshop at NeurIPS [15], the first editions of the Joint Conference in AI Music Creativity [16] and the VPRO Eurovision Style AI Song Contests. [17]

Though there is a lot of interest among academicians and software developers in creating the next set of creative tools using AI for music making, this has not been reflected in the assimilation of AI ideas by musicians of today. Only a handful of musicians, limited to programmers, are using these code based tools in their creative process, as shown by a report of the participation in the VPRO AI Song Contest [18]. As a contrast, in the visual arts, RunwayML [19] is a non-programming environment that has empowered a really large community of AI visual and text-based artists to prototype and realize their ideas without the need for personal accelerated hardware and programming skills by making the technology widely accessible.

Another limitation is the inability to steer and tweak the AI models for personalizing the output within the frameworks available. Current methods are rigid and only allow a single mapping from input parameters to generated music, with the architecture being fixed at the time of training. One potential advantage of AI algorithms as against other computational music systems that follow a set of instructions, is the ability to query higher vs lower probability regions of the representations built. This allows for paradigms of switching between exploitation and exploration modes, a common problem posed in reinforcement learning [20]. Steerable models of AI music systems can therefore be used to not only personalize an interaction but also to explore less likely creative choices by the artists.

*"If you have a computer that comes up with random combinations of musical notes, a human being who has sufficient insight and time could well pick up an idea or two. A gifted artist, on the other hand, might hear the same random compilation and come away with a completely novel idea, one that sparks a totally new form of composition,"*

Margaret Boden  
(2016, interview with IBM)

# 1.1 Objectives

This *Living, Singing AI* system is designed with the following primary objectives:

- **A brainstorming tool for composers to interact and query different kinds of AI models with just voice.**

Centered around an individual’s ideas and aesthetic preferences, the *Living, Singing AI* system provides an egalitarian structure to iteratively interact with a collection of modern AI models simultaneously with just the *voice*. This allows our system to be used without the need for programming or accelerated hardware while enabling the individual to construct novel sonic material generated from a unique vocalized query. This makes it accessible to all kinds of musicians without limitations as to cultural, musical and technological background.

- **A personalized and iteratively proliferating composition object (Organism).**

Furthermore, the system allows the individual to shape the generated material creating a personal, intelligent and evolving model unique to them. The system allows a musician to create a personalized collection of this AI generated material unique to their vocalized input with a collection of tools to select, order and reflect on the generated material that iteratively grows. This bespoke composition object or Organism can consequently be used to generate intelligent mutable compositions in both an individual and a collaborative setting.

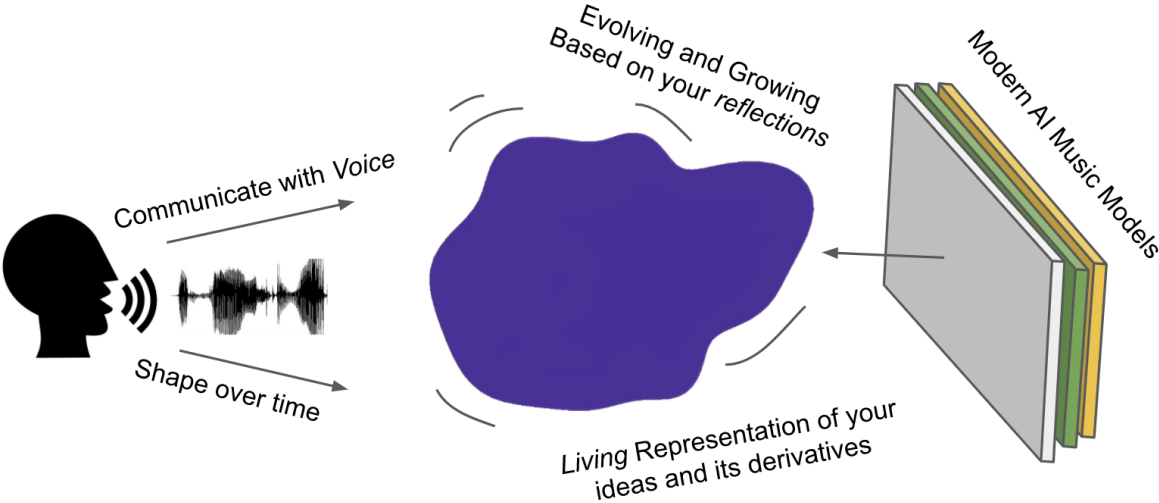


Figure 1.1: Overview of the *Living, Singing AI* system.

## 1.2 Thesis Structure

This thesis comprises six chapters, including this Introduction.

Chapter 2 brings together the background and precedents for this body of work from generative music composition to creative machine learning. In this chapter, we discuss the early adoption of modern AI techniques and softwares in the visual arts and its limitations when applied to music composition. We also survey modern AI Music generation softwares and publicly available technologies with a discussion on the AI Song contest motivating the need for personalizable and accessible tools.

In Chapter 3, we describe a collection of early explorations centered around flexible inputs for generative music systems - specifically voice as the key to making accessible, non-programming AI music composition tools. We also demonstrate the function of a slow-creator, based on computationally slow AI techniques for audio generation, as a bespoke brainstorming tool. These initial experiments lead to a set of design criteria for AI composition systems that motivated the creation of *Living, Singing AI*.

In Chapter 4, we present the *Living, Singing AI* system, its philosophy, key design principles, and its implementation as a brainstorming tool for generating sonic material from one's voice. We describe the underlying AI models, our adaptation to interacting with the voice in an egalitarian framework and the system developed for personalizing the generated sonic material.

In Chapter 5, we discuss a qualitative and quantitative evaluation of the control and range of outputs offered by our brainstorming composition system supported with audio examples from a series of sound collection experiments and informal user studies.

This thesis concludes in Chapter 6 with a reflection on the work presented and a discussion of future directions for developing AI music generation software centered around the creative goals of the individual composer.

## Chapter 2

### Background and Precedents

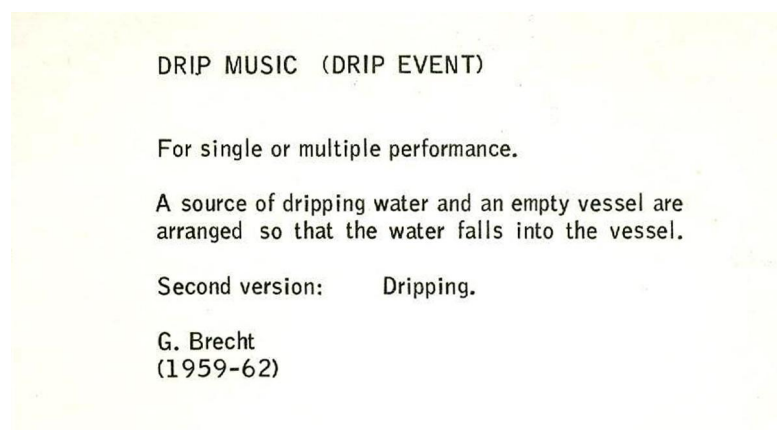
*“With the aid of electronic computers, the composer becomes a sort of pilot: He presses buttons, introduces coordinates and supervises the controls of a cosmic vessel sailing in the space of sound, across sonic constellations and galaxies that he could formerly glimpse only in a distant dream”*

- Iannis Xenakis, *Formalized Music* (1992)

#### 2.1 Algorithms in Music Composition

Ideas about what we now call *algorithms* - essentially, a finite sequence or structure of instructions - originated as early as 900 AD [1]. Algorithmic ways of thinking have allowed composers to work with abstractions of musical concepts in both written and programmed sets of instructions.

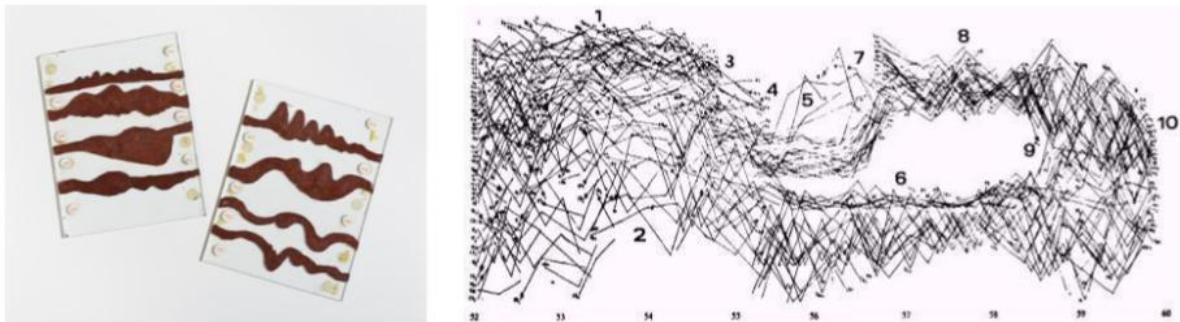
At one extreme, algorithms were treated like musical notations to work with and adapt as vehicles for the composer’s creativity - like works from Wolff, Glass, Stockhausen, Hiller, Ligeti, Xenakis and beyond. Algorithmic instructions were sequenced and stacked just like musical notations on a composer’s staff. Process works were constructed from either a precise or vague description both for note-based and sound-based musics [21]. The term *Process Music* was coined by composer Steve Reich in his manifesto entitled *“Music as a gradual process”* (1968), where he defined it as not the process of composing but rather as pieces of music that are themselves processes.



**Figure 2.1:** Score from George Brecht’s process work - ‘Drip Music’ (1962)



In the early 1960s, Stockhausen created a series of instrumental works as process compositions - *Prozession* (1967), *Kurzwellen* (1968), *Spiral* (1968) to name a few. While these process compositions defined sound events that were feasible, performable and had a definite termination, some algorithmic compositions were unbounded by these constraints. John Cage's *ASLSP* (As Slow as Possible) (1985), originally written for piano and then organ, could theoretically stretch to infinity as the score doesn't specify a tempo, and is intended to play as slowly and as softly as possible in a performance as 'a correspondence between space and time' [22]. A current rendition of the piece at St. Burchardi church in Halberstadt, Germany began in 2001 and is expected to have a duration of 639 years ending in 2640. [23]. In the late 1940s and 50s a radical departure from traditional acoustical instrumentation paved the way for *Acousmatic Music*, a form of electroacoustic music that often existed solely in fixed media audio recordings rather than in live performances. Xenakis was a central figure in bringing algorithmic processes to an electro-acoustic sound-based music. He shared his rigorous and metaphoric expositions in his book *Formalized Music* (1971) [4]. Adopting the affordances of electronic technologies into music composition also allowed for pioneers like Daphne Oram [24] to create new instruments and techniques: *Oramics* - drawing waveforms on glass slides and film strips - was the forerunner to modern sequencing and digital audio workstations (Figure 2.2).



**Figure 2.2:** (a) Daphne Oram's glass waveform slides (1957), (b) Xenakis' graphic score for Pithoprakta (bars 52-60) (1957).

At the other extreme, algorithms have been claimed to be independently intelligent, and in the form of computational agents deemed to be creative [25]. The US League of Automatic Composers (1978-83) and its outgrowth, the network ensemble *The Hub* [26], was an early example of a shared partnership between agents. Each networked computer shared either sonic material or musical processes interacting with each other, letting the network play with or without outside human intervention. Trombone improviser and composer George Lewis' *Voyager* (1980s) was an improvisational software acting as a partner to a human improviser. The *Voyager* software 'listens' via a microphone to Lewis' trombone improvisation and generates a complex response with decisions about melody, harmony, rhythm and silence. [27]

Another example of embodied algorithmic music is *Hatsune Miku* (2007), the Japanese anthropomorphic pop star that uses the Vocaloid singing voice synthesis algorithms [28] from Yamaha. In 2020, the giant talent agency CAA signed *Lil Miquela*, one of TIME magazine's 25 most influential people on the internet as well as being a fictional computer-generated digital musician, not a real person.

These historic examples show that algorithmic music can perform diverse social functions from a passive tool for generating, structuring and performing old and new paradigms of music to an active agent as an instrument, partner or even the main performer.

Algorithms in principle can contribute to all stages of music making, what historically has been a highly differentiated series of activities: composition, performance, improvisation, spatialization, recording, editing, mixing and mastering.

### 2.1.1 Generative Composition Tools

Musikalisches Würfelspiel [29] (German for dice music) is one of the earliest examples of a tool to aid in music composition. In the mid 1700s, composers could roll dice to pick bars of music from a table, socket them into tile-based rows and create complete forms of music.

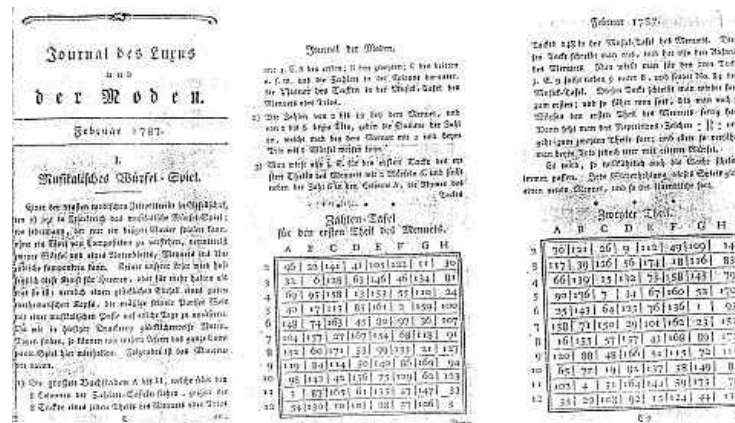
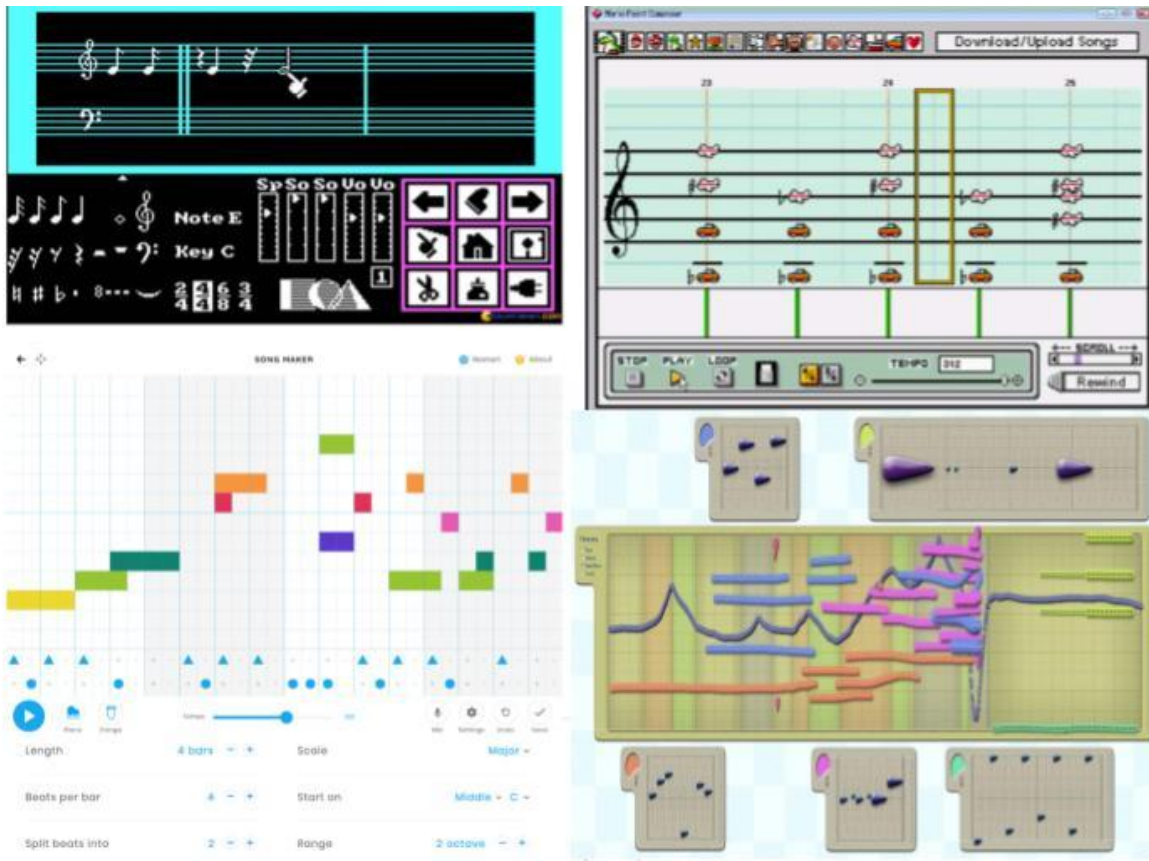


Figure 2.3: Musikalisches Würfelspiel (1757) 1st edition proposed

This popular music making *game* was accessible to amateurs and all results would be permutations of smaller units of pre-written music. It was an extension of similar composition ideas from C. P. E. Bach's *Einfall, einen doppelten Contrapunct in der Octave von sechs Tacten zu machen, ohne die Regeln davon zu wissen* (German for "A method for making six bars of double counterpoint at the octave without knowing the rules") (1758). Automated techniques (e.g. through randomness) have been used, for example, by Mozart in "Dice Music," by Cage in "Reunion," and by Messiaen in "Mode de valeurs et d'intensités." The created compositions via such algorithms still needed to be performed on an instrument in order to auralize the music and judge the quality of one's creations.

Following early digital synthesis techniques, *Will Harvey's Music Construction Set* (1983) took the same ideas of jigsaw-style sequencing to individual notes. This paved the way for a style of music composition systems that allowed one to sequence individual notes to generate longer pieces. Mario-Paint composer (1992), Hyperscore (2002; 2021) and Google Song Maker (2018), shown in Figure 2.4, are a collection of music composition systems demonstrating the popularity and continued relevance of such composition softwares. This allowed for a wider variety of musical outputs, though still constrained by fixing a

pre-defined key, tempo or harmonic vocabulary, to keep the outputs within a safe musical space of possibilities. These were therefore intentionally intended for amateur composers.



**Figure 2.4:** Music Construction Set, 1984 (top-left), Mario Paint composer (1992), (top-right), Google Song Maker, (bottom-left) and Hyperscore, 2002/2021 (bottom-right).

Generative composition systems like these capture representations of music through a rule-based system. Lejaren Hiller and Leonard Isaacson's *Illiad Suite for String Quartet* (1957) [30] is the earliest example of such a rule based generative system credited as the first work completely written by artificial intelligence. This system used various stochastic models building markov chains for simple diatonic melodies and fixed rules to generate four voice harmony on a 1952, ILLIAC-I computer.

Experiments in Musical Intelligence (EMI) in 1980 by David Cope also used the strategy of recombining instructions captured as a representation of classical style composers like Bach, Beethoven, Chopin etc. David Cope described his initial experiments as *uninteresting and unsatisfying* but he recognized and demonstrated the potential of algorithms as tools to realize very *human ideas*.

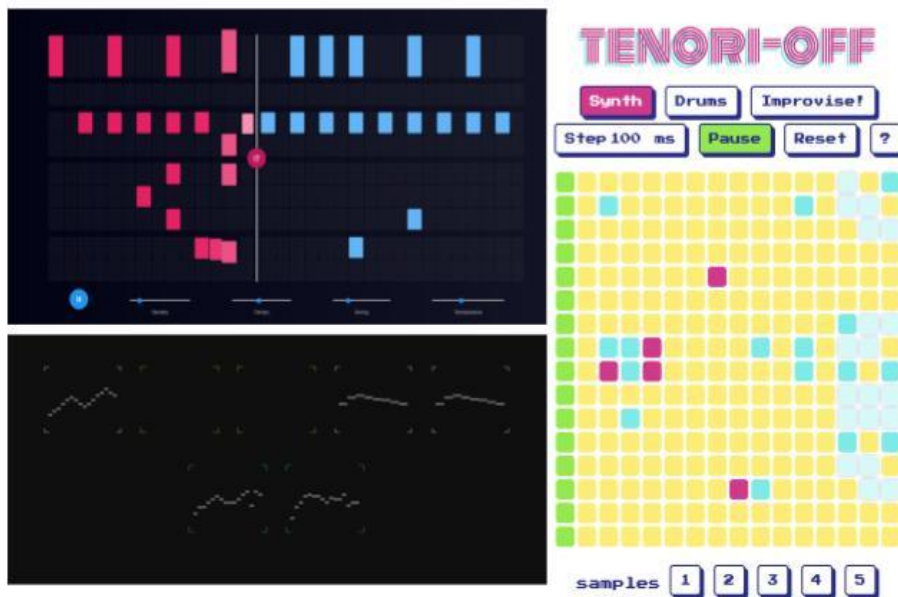
*“The works have delighted, angered, provoked, and terrified those who have heard them. I do not believe that the composers and audiences of the future will have the same reactions. Ultimately, the computer is just a tool with which we extend our minds. The music our algorithms compose are just as much ours as the music created by the greatest of our personal human inspirations” - David Cope on EMI [31]*

It builds a simple grammar where repeated musical patterns observed in an input dataset are stored in a dictionary, followed by recombinations producing variations [32]. David Cope’s work can be considered a kind of algorithmic pastiche - an imitation of a certain style or period.

The newer sounds of electronic music, both hardware (modular synthesizers) and new synthesis techniques (FM synthesis, 1974, Granular synthesis, 1971) did not have a precedent of a past canon of styles to imitate. It allowed for extended experimentation of the affordances of such *Generative Systems*. The word ‘*generative music*’ was popularized by electronic musician Brian Eno through his experiments of infinitely running, non-repeating ambient work. By repeating tape loop cycles of different lengths, Brian Eno created his long form generative systems like *Ambient 1: Music for Airports* (1978).

*“From now on there are three alternatives: live music, recorded music and generative music. Generative music enjoys some of the benefits of both its ancestors. Like live music, it is always different. Like recorded music, it is free of time-and-place limitations — you can hear it when you want and where you want.” — From “Generative Music” in A Year With Swollen Appendices by Brian Eno (1996)*

Newer representations of music built through Variational Autoencoders (VAE), Recurrent Neural Networks (RNN) and also Transformer models (Introduced in detail in Section 2.3) have now been used to demonstrate the same kinds of early generative music systems - (a) imitating a specific domain style and (b) infinitely permuting on a fixed musical vocabulary. These also suffer from similar limitations of producing a narrow range of similar sounding outputs and giving minimal control for personalization. These are therefore best used as short musical games for amateurs.



**Figure 2.5:** *Neural Drum Machine* - DrumsRNN (top left), *Sornting* - Music VAE-based web game (bottom-left), *Tenori-Off* - ImprovRNN sequencer (right) - implemented as browser applications.

There are now many programming languages and environments designed specifically for the expression of algorithmic music, with the classic Music-N (CSound), Lisp (SuperCollider) and Patcher languages (Max/MSP, PureData) joined by Gibber, Sonic Pi, Tidal Cycles, ORCA and many more. Though limited to composers with programming skills and a hacking aesthetic, a culture of composing and even performing live with algorithms has emerged centered around creators. Some of these communities like TOPLAP, Live Algorithms in Music Group and NowNetArts are paving the way for a new generation of algorithmic composers across the world.



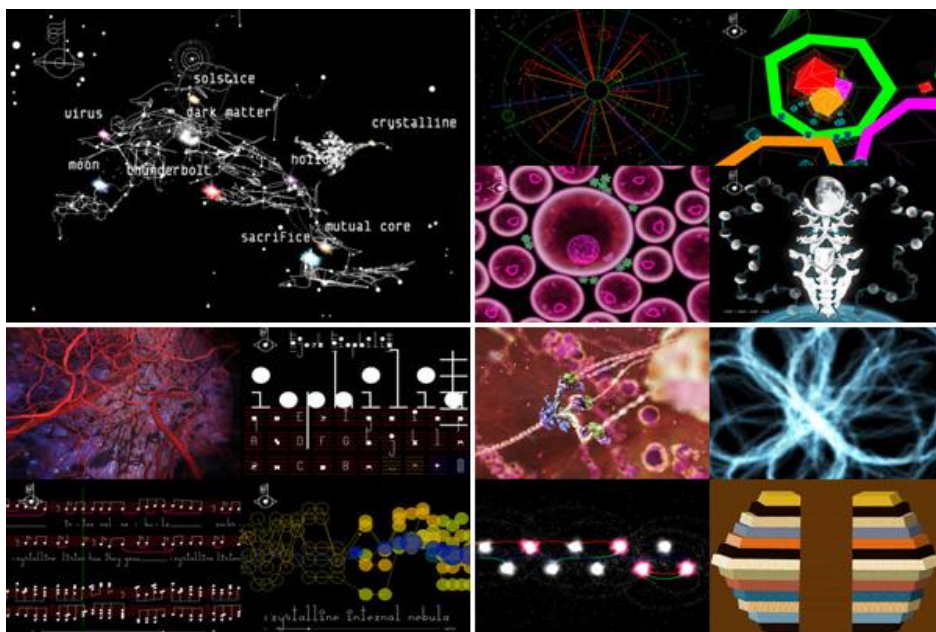
**Figure 2.6:** Alex Mclean, creator of the programming language Tidal Cycles performing at an Algorave

A general approach for such algorithmic composers involves presenting the algorithm as an instrument with a focus on performance. Performers write sequential lines of code in an esoteric programming language that generates music in real-time. The coding performance is projected onto a larger screen for the audience as a way to share the intent of the performer [33].

## 2.1.2 Interactive Compositions

Algorithms in music compositions can also be used to create a *reactive* composition, which responds to environmental input or an *interactive* composition, which listeners interact with directly to influence the music. Many forms of procedural music have been widely used in video game music [34], which is influenced indirectly by a state of the gameplay and actions of the player.

Björk's 2011 album *Biophilia* [35] explored the concept of a mutable composition for intentional participation by the listener in the form of mini-game applications. Each song was accompanied by a mobile app that allowed interactions with the main melodies, song sections etc., changing the composition into a few pre-determined forms.



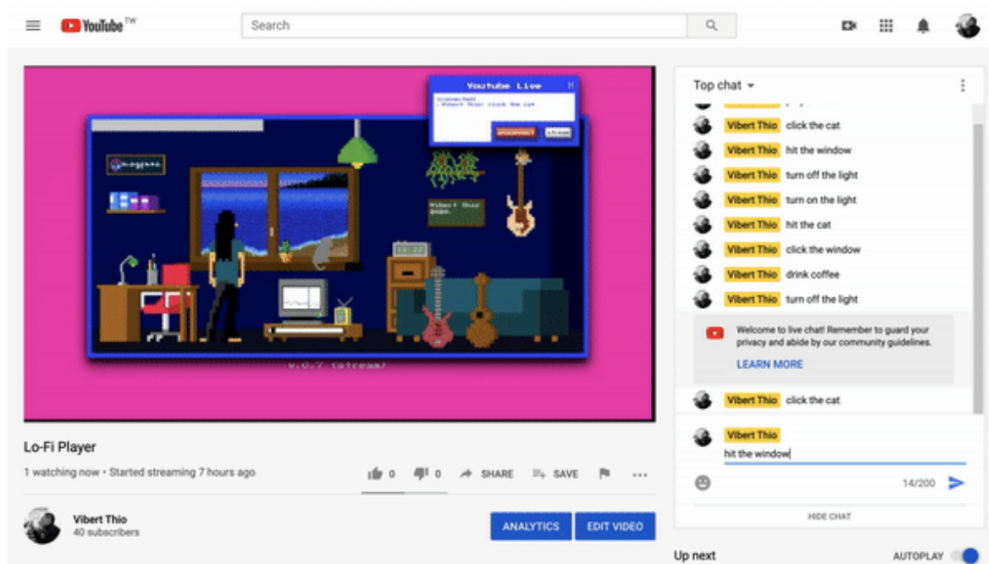
**Figure 2.7:** Biophilia 2011 - Björk's album accompanied by a suite of mobile apps for interacting with each song.

More recently in 2021, Google released a viral interactive four voice composition system called the Blob Opera. Dragging the blobs up and down changes the pitch of one of the voices while the others harmonize using a Convolution Neural Network (CNN) to generate the voice waveforms. It allows you to step through an existing classical opera composition. Though it doesn't allow you to make your own compositions, a simple control resulting in a complicated human singing voice aided its popularity. In the past most generative games used really simple sound sources like MIDI piano.



**Figure 2.8:** Google’s Blob Opera, Screenshot from the browser application.  
(<https://gacembed.withgoogle.com/blob-opera#/>)

The Magenta project by Google, started in 2016 as an open-source development of machine learning as a tool in the creative process. An example, created by Magenta, of an AI composition being generated on the fly with inputs from listeners was the Lo-Fi player (Sept 2020) which used their open sourced MusicVAE and MelodyRNN models, both lightweight generative models that generate multi track MIDI sequences. One could interact with a live streaming version of the system by entering commands in the chat like ‘change the melody’ or ‘change instrument’. This implements a very simple resampling of a new random melody from the model. Similar to *Blob Opera*, this system does not allow an individual to personalize the output but provides a small sense of control.



**Figure 2.9:** Lo-Fi Player from Magenta implemented as a Youtube livestream taking inputs from live chat.

These examples, though limited, show the potential affordances for AI music compositions as morphable and interactive both from the perspective of the composer’s choices and the active listener’s preferences.

## 2.2 Creative Machine Learning

The ability of Generative Artificial Intelligence algorithms to capture a latent representation of the dataset that it is trained on has enabled a few new paradigms for what it means to be an *AI artist*. This can be observed in the creative machine learning community especially in text and visual arts that have been early adopters of this technology.

### 2.2.1 Latent Space Explorer *Artist*

Ian Goodfellow in 2014 created the Generative Adversarial Network (GAN) [36] architecture that generates synthetic data from a latent representation of a training dataset using two competing neural networks models - a discriminator and a generator in a zero sum game. Though the architecture and its many variants can be trained on any customized dataset, models trained on standard large image datasets (example ImageNET) have become popular. Since training such models on completely new dataset requires a lot of GPU computational resources, artists have been exploring the freely available pre-trained models. Artist Mario Klingemann demonstrates a novel approach to exploring these latent spaces like a cartographer, creating journeys through the latent space as a new form of exploratory art.



**Figure 2.10:** AI Artist Mario Klingemann's twitter (dated 15th Nov 2018). Describes latent space exploration as map making.

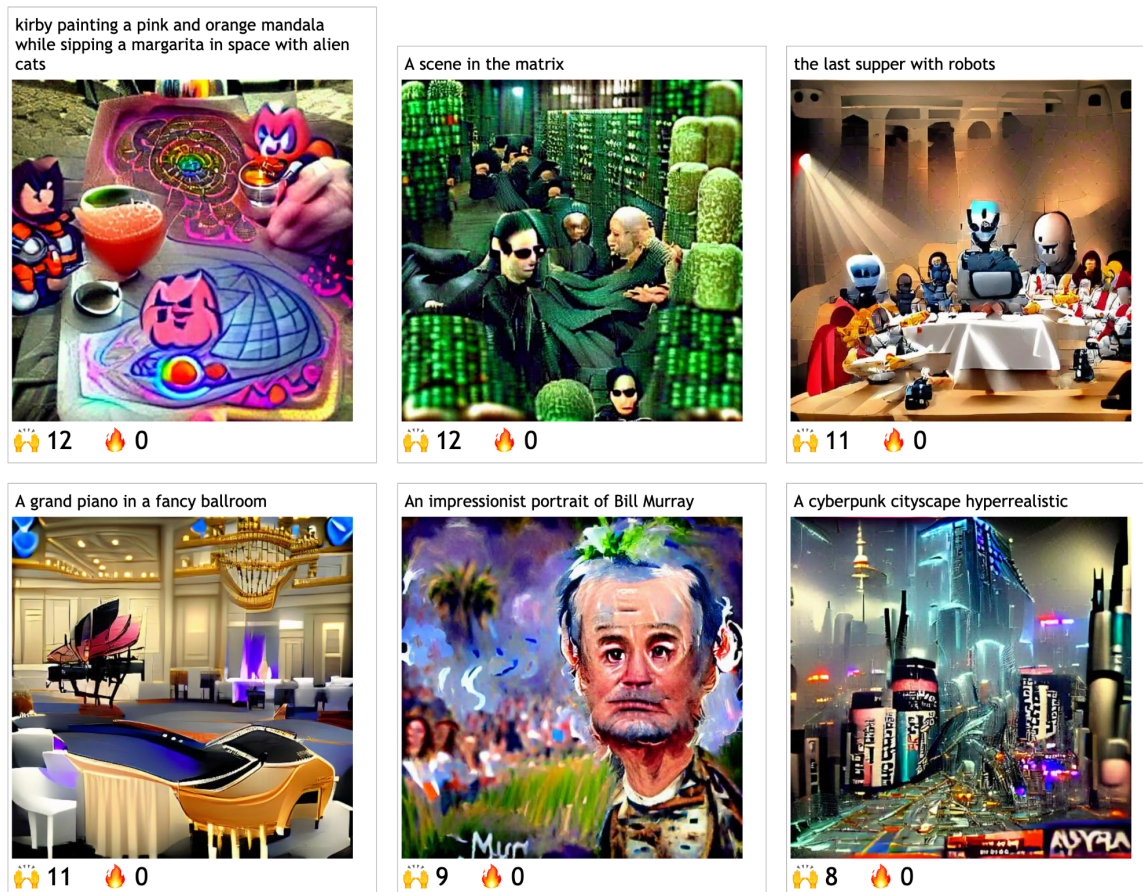
### 2.2.2 Prompt Creator *Artist*

Models that connect text prompts as a way of querying the latent representations like the transformer models in the case of GPT-3 (a language transformer model), CLIP, DALL-E (OpenAI's text to image representations), all demonstrate a new skill of designing the perfect queries. This has resulted in an opportunity to be creative with text prompts. The text prompt is converted into a sequence of tokens



that uniquely sample a point in the underlying representations, thereby acting as a hook into the latent space.

Here is an example of a wide range of creative prompts and results from the CLIP + VQVAE models connecting text and images, proposed in Jan 2021. [37].



**Figure 2.11:** Abraham-AI a visual arts generating Decentralized Autonomous Organization (DAO) with examples produced by creative human prompts. <https://abraham.ai/create> (accessed Aug 10, 2021)

The limitation of being able to interact with a generative model only through such prompt queries has been found to reveal biases in the models [38]. OpenAI’s DALL-E model revealed that adding “...unreal engine” at the end of a prompt resulted in more polished 3D render style results since the examples in the training set had a strong correlation with those labels. Unreal Engine is a game development engine which renders high quality images. This effect of engineering the prompt for better outputs is also popularly being referred to as the “unreal engine trick.”

### 2.2.3 Curating Dataset *Artist*:

Another new form of AI art aesthetic is when an artist uses commonly available AI models but trains it on personal and thematically curated datasets. The artist collects a dataset of images or text examples

from a narrow theme such that patterns in the data are consistent and therefore easy to capture as latent representations. Here we show examples of - Harshit Agarwal (2018) (Figure 2.12), who uses a curated dataset of images of human surgical dissections to create novel machine generated dissection images and Memo Akten, who uses a self collected dataset of cloud photographs.



**Figure 2.12:** (top) Harshit Agarwal's *The Anatomy Lesson of Dr. Algorithm* (2018), generative art from a curated dataset of 60,000 images of human surgical dissections. (bottom) Memo Akten's *Learning to See: Gloomy Day* (2017), generative art from a dataset of clouds.

#### 2.2.4 Accessibility of *AI Creative Tools*

This early adoption and experimentation of AI ideas in the visual media can be attributed to several reasons, an important one being accessibility to the tools.

Apart from open-source code shared through Github, pre-trained models have been made available through ventures like Hugging Face that host massive transformer models. (Examples include DALL-E mini, GPT etc.). Browser-based and standalone software tools for interacting with the models without the need to write code allows non-programming visual artists to quickly engage with newly released AI models.

RunwayML is one such platform created in 2018 to democratize and share models amongst the AI artist community without the barrier of programming skills. Simple drag-and-drop interfaces enable quick

prototyping. It also takes advantage of cloud based GPU services that do not require expensive computational hardware requirements from the users.

An ecosystem of accessible software, shareable models, online MOOCs [39] on creative machine learning, and a growing marketplace through dedicated AI art galleries [40], museums and Non-fungible token (NFT) platforms [41] suggests the undeniable growing impact of an evolving AI art community in the current decade.

*“Because these systems are so complex, you cannot really say, ‘Okay, I change this number and that thing will work. It’s really turning knobs here and turning knobs there — maybe like with analog synthesizers or cooking. You learn by experience that it probably goes in this direction but then there could always be something really strange happening around the corner. And of course that’s what you’re hoping for ... that you end up in a space that you didn’t even know was there.”*

Mario Klingemann, AI Artist [42]

Mario Klingemann’s quoted experience and excitement of exploring this new form of AI art directly reflects Xenakis’ comments, quoted at the beginning of the chapter, about being an explorer pilot of a cosmic vessel pressing buttons and turning knobs.

So far we have seen examples of these buttons and knobs, or more generally, *inputs* to AI Music generation systems situating at two extremes:

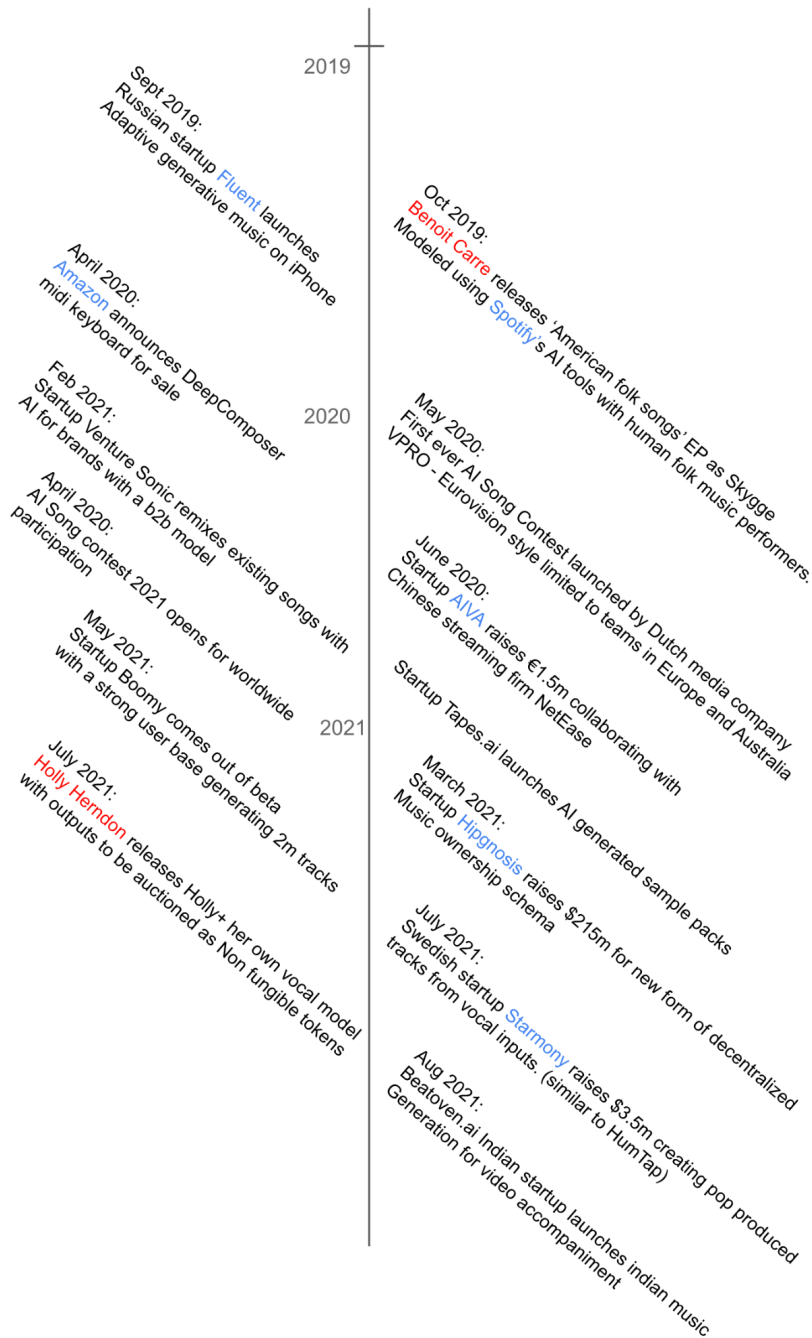
- (a) Very High level - Limited input control - choosing a mood, genre, key, tempo, etc.
- (b) Very Low level - Software frameworks that have to be programmed / hacked by the artist themselves from the ground up.

## 2.3 Artificial Intelligence in Music: Overview

### 2.3.1 Timeline of AI Music Entities

In the last few years, development of synthesis techniques for audio and modeling music have led to a surge of interest and activity from small start ups, individual technology-artists and also bigger technology companies like Google, OpenAI and Facebook. Some of these key events in the timeline have been summarized in Figure 2.13.





**Figure 2.13:** AI Music industry timeline, (blue marks businesses, red marks artists)

This timeline was compiled from multiple reports by *music:ally* a global music technology consulting group [43] that produces editorial insights and market research reports covering the changing music industry. We can observe that most startups like Amper, Melodrive etc. have focussed on targeting AI music making towards brands, enterprises and video game industry for quick scalability rather than for individual creators. We also notice a few companies collaborating with artists but most notable artists

have worked independently programming their own AI music models and systems. While artists like Taryn Southern, YACHT, Flaming Lips have collaborated with companies to adapt an existing AI software for a minimal role in the composition process; more interesting aesthetic results were obtained when artists curated their own datasets and built their own models.

Artist collective *Dadabots* went with the approach that the ‘*AI algorithm is the composition*’ [44] by training a Sample RNN model on death metal music generating a non-stop stream directly to YouTube. Artist Holly Herndon on the other hand, trained voice models on a vocal ensemble and her own voice for her album *PROTO*. Her voice clone model called *SPAWN* was considered an *equal partner* in the album creation process and the generated sonic material was carefully curated and produced for each song by the human partners to create the finished album. The model is a variation of the SampleRNN architecture similar to *Dadabots*, but is trained on studio sessions of Holly’s voice samples recorded to create a novel dataset. Holly has also been experimenting with creating a decentralized autonomous organization (DAO) for her voice clone model (*Holly+*) that can be used by other creators to create music with and further sell. [45]

It is clear from these examples that AI algorithms for music generation can push the culture of music composition, ownership and consumption but only when accessible to a diverse group of individual music makers. Each individual brings a unique set of creative perspectives and aesthetic preferences informed by their cultural background. Without this diversity, AI music would be boxed into narrow definitions of style, genre and function defined by the select few who have access to it.

*“When the computer is doing lot of the heavy lifting for us, it’s really allowing us to be more human together”*

- Holly Herndon  
AI music artist [46],  
(2019) interview for New York magazine’s Future issue

## 2.3.2 Timeline of Generative AI Music Software

### A. Early Methods:

Historically, music has been considered at a symbolic level, where symbolic information like pitch, duration, loudness levels and MIDI specifications are the result of choices made by a composer. This originates from a formalism of music as a system of sounds, intervals and rhythms that dates back to Pythagoras, Ptolemy and Plato, who considered music as inseparable from numbers. Automated composition systems naturally modeled music as such discrete symbols but were also capable of modelling analog and discretely sampled representations of continuous sound.

Computer-generated automated compositions can be categorized into four main methods:

### *Stochastic Methods:*

This describes systems which use sequences of jointly distributed random variables to control specific decisions. Markov chains were a very popular method in the early years of algorithmic composition [47] because of their low complexity. Though such Markov chains were limited in capturing just local statistical similarities, it is still widely used for specialized tasks like jazz chord progressions [48], musical transitions between melodic sequences [49] and even by Pachet, et al [50] for their real-time melody generation and the *Continuator (2002)* [51], a machine listening interactive system.

### *Rule based Methods:*

This comprises systems which use a strict grammar and a set of rules to control decisions. One of the earliest automated compositions, *Illiad Suite (1958)* [30] was generated using such rule systems. In broad terms, a grammar in this context may be defined as a set of rules to expand high-level symbols into a more detailed sequence of symbols representing elements of formal languages [52]. David Cope's "Experiments in Musical Intelligence" system [53] is also an example which learns the *style* of a composer given a certain number of pieces by storing repeated patterns in a dictionary and building a grammar with rules on re-combining the stored patterns. Rule-based systems like learning decision trees are also easier to interpret semantically meaningful.

### *Evolutionary Methods:*

Evolutionary methods are population based meta heuristic optimization algorithms, so in the case of music a population of musical fragments are computationally selected, recombined and mutated with a fitness function. This set of algorithms was also popularly used for generative music systems like Al Biles' *GenJam* system for jazz solos (1993) [54], Rodney Waschka II *Gen Dash* [55], Horowitz [56] on genetic algorithms for rhythm and also the EuroGP Song Contest in 2004 [57]. Santos et al. [58] provides a survey of popular trends in evolutionary methods for computer-generated music systems.

### *Artificial Intelligence Methods:*

This is the category of systems that capture their own rules in supervised, unsupervised or reinforced methods in essence to "learn." A few early examples are Ebcioglu (1988) CHORAL [59], a backtracking expert system for the harmonization of chorales in the style of J.S.Bach, Todd's (1989) [60] feed-forward neural network for melody generation and Toiviainen's (1995) [61] neural network for Jazz improvisation. Toiviainen [62] identified the limitation of those simple neural network models that failed at capturing high level features of music related to phrasing or tonal functions.

This is the approach most significant to our work and some of the more modern approaches are described in greater detail here.

## **B. Data driven methods:**

As opposed to hand-crafted methods like the rule-based and grammar-based techniques mentioned earlier, machine learning and specifically deep learning (DL) methods that involve a training step on datasets have a form of generality. The method itself is agnostic to the corpus of data used to train the models and could theoretically be used on any collection (style / genre) of music. As stated in Fiebrink and Caramiaux [63] some benefits of such data driven models are:

- (i) they can make creation feasible when the desired application is too complex to be described by analytical formulations or manual brute force design; and
- (ii) learning algorithms are often less brittle than manually designed rule sets and learned rules are more likely to generalize accurately to new contexts in which inputs may change.

For such data-driven methods that rely on large collections of music data, an important problem is curation of the datasets. Since music is often a commodity unlike texts and images which are available under free licenses to be scraped from the web, collecting such datasets is an important challenge.

## **C. Datasets (openly available):**

One way this challenge has been tackled in academic contexts is by creating a shareable form of metadata for the musical information rather than audio content itself. This metadata represents extracted signal characteristics from which the audio signal cannot be reconstructed. Million Song Dataset [64] and AcousticBrainz [65] are such metadata collections of extracted audio content features from music. Free Music Archive [66] and Freesound [67] are examples of royalty free crowd contributed audio collections.

While it is important for the academic community to build better models by cross validation over commonly shared music datasets, the goals from a creative motivation are different. Often composers wish to curate their own sound collections and create models that are not built on a past canon of work directly when not creating *pastiche art*. This further motivates the need for making models that are personalizable.



### D. Popular Open-Sourced Models:

Here is a timeline of open-sourced methods, models and a few softwares APIs that are available in the public domain (2016 onwards):

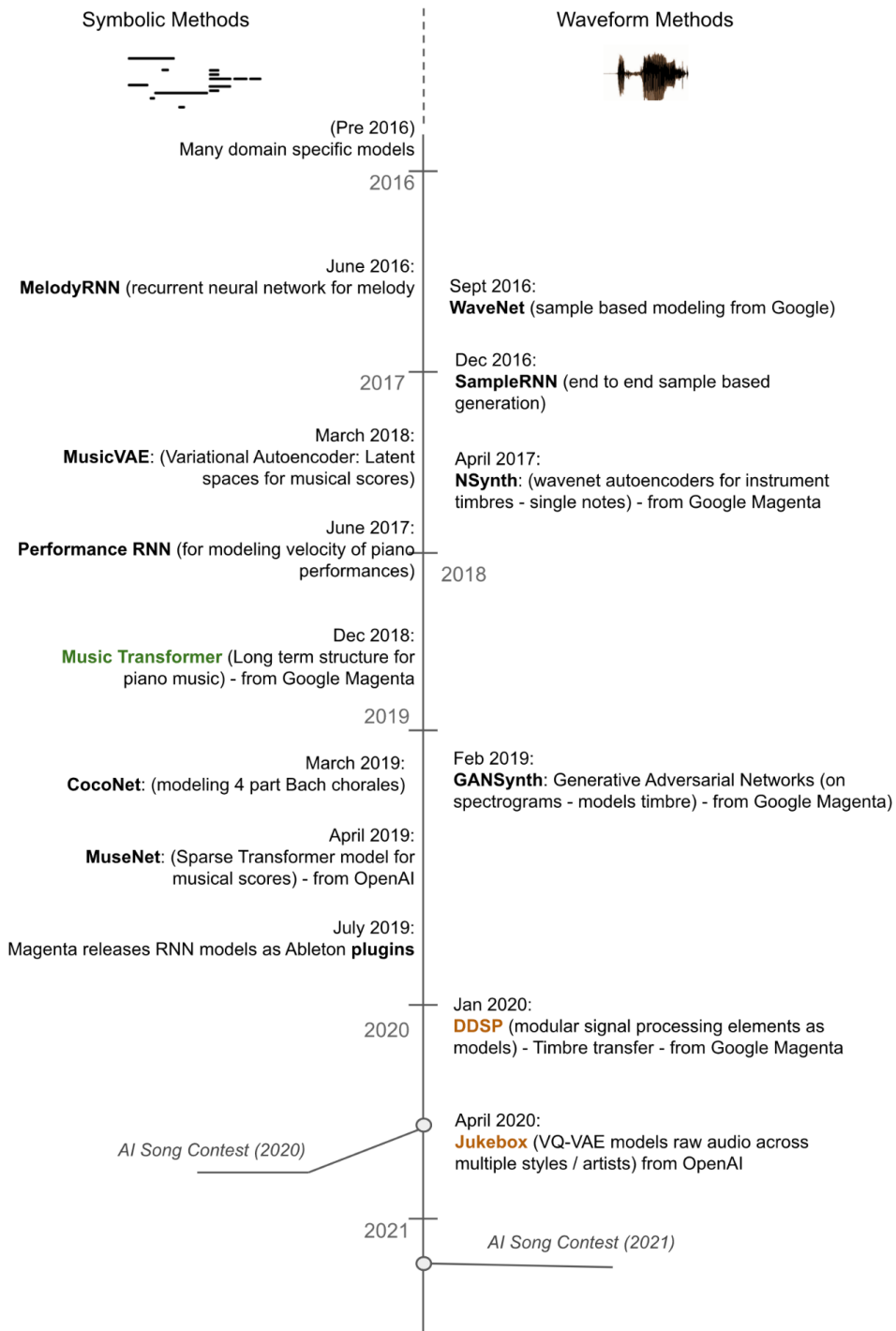


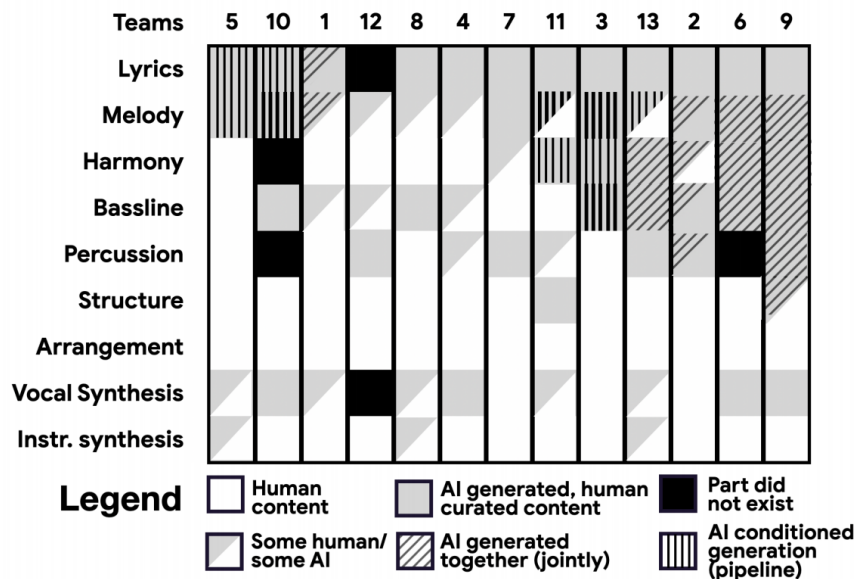
Figure 2.14: Open source AI Music Software timeline.

Since most deployed AI music services are built for enterprises with the goal of fast and frequent music generation, the focus is not on the interests of individual composers. The tools are often very limited in the ways they can be customized. This limitation was highlighted in the first ever AI Song Contest [17] and its accompanying Joint conference in AI Music Creativity [16]. The AI Song Contest was the first time a panel with a common evaluation strategy was used to invite artists and engineers to create AI Music with the focus on the music rather than the technology. The context for the competition was limited to *Eurovision*-style pop music with a provided dataset of past *Eurovision* winning songs. In the next section we point out a quick summary of the events in its last two editions and the observed limitations of current AI Music tools found in practice.

## 2.4 AI Song Contest

In 2020, the first ever AI Song contest was organized by Dutch public broadcaster VPRO with 13 participants. Its limitation of participation from only Eurovision countries was dropped in the next edition in 2021, which had 38 participating teams spread across the world including from the U.S., Nepal, South Korea, etc.

The competition loosely defined what role AI had to play in the process of the song intentionally encouraging multiple paradigms of partnership, authorship and musicianship that are challenged and extended in the context of AI. Here is a description of the paradigms and AI technologies used by the teams reported in Huang, et al [68].



**Figure 2.15:** Team numbers of each of the 13 participating teams and the AI-human partnership paradigm used for each independent component of the song indicated. Figure referenced from [68].

We observe that text-based AI for generating lyrics was the easiest to implement and consequently most common across all participating teams. Symbolic music modeling for generating melodic, bassline and drum sequences was the next most popular. Google's Magenta in 2019 had provided some of their RNN models, such as Electron and MAX MSP Plugins, which greatly impacted their popularity in this contest. This ease of use and familiarity of plugin environments in Ableton was observed by its popularity as an accessible technology amongst participating teams.

### 2.4.1 Common Strategies for the AI Song Contest

- **Generate and Select:** A common approach was to use models to generate a large volume of samples and sequences to choose from. This would be followed by a human curation step for handpicking the most appealing.
- **Generate Vocabulary not Structure:** Most models had the capability to generate short segments of samples and sequences rather than long-term hierarchies. Longer structural ideas like song structure, themes etc. were performed without the aid of AI tools.
- **Active co-creation:** Though less common, some teams co-created with the AI models, where the model outputs influenced human composing followed by a second stage of model generation with more refined parameters. Models were not inherently designed to incorporate iterative exploration. Such a strategy seemed natural to the way human musicians would collaborate with each other.
- **Inspiration:** Generated material from the AI models worked very well as inspiration for identifying themes and motifs by human musicians. An example of this methodology was one approach where Sample RNN was used to generate babbling human singing voice samples as a first step. This was followed by a singer identifying sections and phrases of their liking and then re-recording human singing renditions of the same.

### 2.4.2 Limitations

Unlike in previous cases where musicians built their own models and worked with AI architectures from the ground up with a creative goal in mind, this year's competition saw composers trying to be creative with existing techniques and tools. Some of the limitations of the existing models when approached from the perspective of the composer were apparent, such as:

- *Low aesthetic quality:*  
Many models simplify the audio data stream by downsampling the standard 44KHz/44.1KHz to 8KHz/16KHz as a pre-processing step. This is done to help the model capture longer time representations. Other models that work as filters on spectrograms also disregard the phase information and use a downsampled magnitude spectrum. These methods help make the computations practical but sacrifice the quality of audio produced. Downsampling artifacts,

phase coherence artifacts, in addition to lack of longer time consistencies produce outputs that are generally lower in quality than regular digital synthesis methods. This results in a low aesthetic quality of sounds created by AI algorithms.

- *Not directly steerable:*  
While some models work with a priming MIDI sequence to create harmony, continuations and extrapolations for, most models give very few controls to steer the model. The architecture is designed in a traditional Machine Learning method of passing parameters to obtain an output. Iterating requires restarting the algorithm with different parameters. While some popular AI tools like AIVA, offer even fewer parameters to control, like a fixed key, tempo, mood and genre. Trading control for powerful generalized mapping is not ideal in the creative task of composition. The provided controls are very limiting and do not allow for personalizable outputs.
- *No contextual and structural knowledge:*  
Unlike human collaborators, AI algorithms do not have a context of the instrumentation, cultural background and even larger goals of the human composer. Each generated material from these algorithms is isolated and out of context. Of course this leaves the task of creating context with the human, but the lack of constraining the system by context creates a needle-in-the-haystack problem.
- *Training data bias:*  
If the datasets of the models are trained on music of a certain style, aesthetic or musical domain, the model assumes all inputs to fall in that category. This fails when the domain constraints don't match that of the input.
- *Evaluation limitations:*  
While the AI song contest used a two part expert panel and audience voting to select a winner, evaluating the role of AI as a successful tool for the composer was not evaluated. Comparing multiple AI algorithms and methodologies in the workflow of a composer is lacking.
- *Plagiarism:*  
A few teams created plagiarism checks to identify if material produced from the AI models was plagiarised from the training set. This is a broader copyright issue, highlighted in Section 6 of this thesis. AI algorithms might even be helpful in identifying certain kinds of plagiarism in the future.

Rebecca Fiebrink et al [69] produced a collaborative study interviewing 7 composers who were invited to use Wekinator [70], a real time interactive Machine Learning tool for creating mapped interfaces, and found the principles that composers valued. Here is a selection of the principles reported by composers from the study that are reflective in the context of the AI Song contest.

*“Providing access to surprise and discovery.”*

*“Balancing discovery with control and constraint”*

*“An invitation to play”*

*“Attention to accessibility in a creative paradigm”*

While there were many limitations faced by composers in trying to use existing AI technology, it also provided many opportunities challenging humans in the loop to be creative in directions they ordinarily wouldn't have. Certain samples originating from the AI models became the main themes which were expanded upon by human performances and re-interpretations in the submissions. Bill Buxton [71] describes a “flare and focus” cycle of ideating, exploring those ideas, selecting ideas and iterating as the central aspect of a creative process. Different stages of the creative process involve a balance between exploration and exploitation strategies. The AI models offered more as undirected exploration tools than as exploitation tools in their current state.

### **Model tuning as Performance:**

Traditional generative AI models capture representations from the training data into a latent space which can be sampled randomly. The latent space might be discrete (like autoencoders) or continuous (like variational autoencoders). Sampling from such latent spaces is often similar to looking up in a dictionary of possibilities. A more flexible way of querying the model and sampling would allow an individual to take up a more performative role. This would be more akin to treating the algorithm as an instrument. Even when algorithms are slow to sample and therefore cannot be treated as a classical realtime instrument, they would allow a form of slow creation akin to early music creation by Max Matthews [72] using punch cards on an IBM mainframe.

## **2.5 Voice as Input**

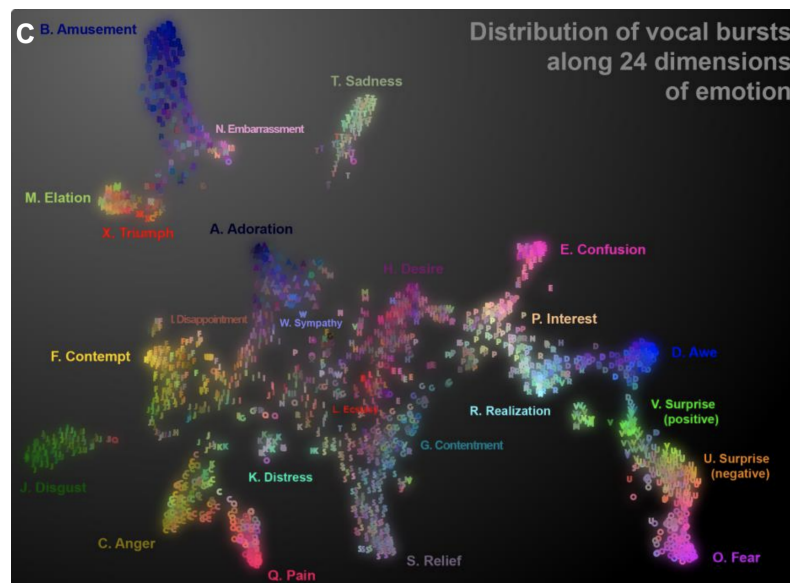
Using our voice and language through speech is a common way we interact with technology today. Early voice-to-text systems were only available as a dictation tool (Dragon Dictate 1990 [73]) but are now ubiquitous in the age of the Internet Of Things (IoT). In her panel talk at CHI 2017 [74], Pattie Maes observes that human computer interactions have a trend towards shifting from being more mechanical interactions like punch cards etc. to more human interactions like language, voice, touch etc. Our devices become a more seamless extension of ourselves that augment and assist in our intended activities. This could also be reflected in the way we interface with our AI Music Algorithms through sound, music and voice rather than lines of programming and creating machine readable prompts.

In our system, voice is used as an input to generate sonic material which is acoustically related to the input. In other words the input is treated as non-verbal sound with acoustic properties like loudness, pitch, harmonic, percussive information, transient onsets etc.

## 2.5.1 Query by Humming:

Using the voice as a way to extract pitch information was first introduced as a type of Music Information Retrieval question. In its very first version [75], a vocalized query of a hummed melody was used to retrieve a matching song from a library. This medium of interacting with technology using our voice was intuitive and quickly became a standardized annual challenge at the Music Information Retrieval Exchange (MIREX) from 2006 onwards with a large body of research. Query by humming techniques have since been implemented in audio search on mobile devices by Soundhound and Midomi (2005) and more recently on Google Search (launched in 2020) as “hum to search”.

Query by Vocalization (QBV) systems use vocal mimicry, typically non-speech utterances intended as a direct acoustic representation of a melodic contour. Such non-speech vocalizations also span a wide range of intent. Cowen et. al [76] show the capacity of short vocalized bursts visualized by their acoustic representations covering a range of 24 emotional intents. In musical contexts, non-verbal vocal mimicry is an ancient art form used in theater and foleying. “Imitation of Birds” is one of the earliest known physical recordings of vocal mimicry from Chennai, India in the early 1900s [77] where a vocal mimicry artist dramatizes a scene from the city - birds chirping, imitation of a passing train.



**Figure 2.16:** Cowen et. al. [76] demonstrate 24 emotions mapped from short Vocalized queries (QBV)

The *voice* and *vocalized non verbal utterances* have been used as control inputs by a few different kinds of interfaces for musical expression like music therapy [78], Vocal Tangible Interfaces [79] and interactive tactile vibrations through *Vocal Vibrations* [80]. HumTap [81] - a popular mobile-based composition game, allows amateurs to hum melodies that are converted into preset musical structures. Vochlea [82] on the other hand is a hardware interface that acts as a voice to MIDI controller in a microphone form. The demonstrated range of *non-verbal vocalized* utterances and the power of ‘*voice*’ as a form of input control in different contexts motivates a similar way of flexibly interacting with modern generative AI algorithms.

## Chapter 3

### Initial Explorations

*“It’s possible that our grandchildren will look at us in wonder and say, ‘You mean you used to listen to exactly the same thing over and over again?’”*

- Brian Eno  
(Notes on Generative Music, 1997)

In this section we explore a selection of previous projects that helped to motivate the broad design goals of the *Living, Singing AI* system and also develop specific techniques and methodologies for the proposed AI musical composition system. For each project, we first describe the concept followed by the implementation of the system. Finally we provide an analysis of the challenges, learnings and targeted directions we incorporated into the design of our system. At the end we summarize the key learnings and ways in which they connect to the primary work.

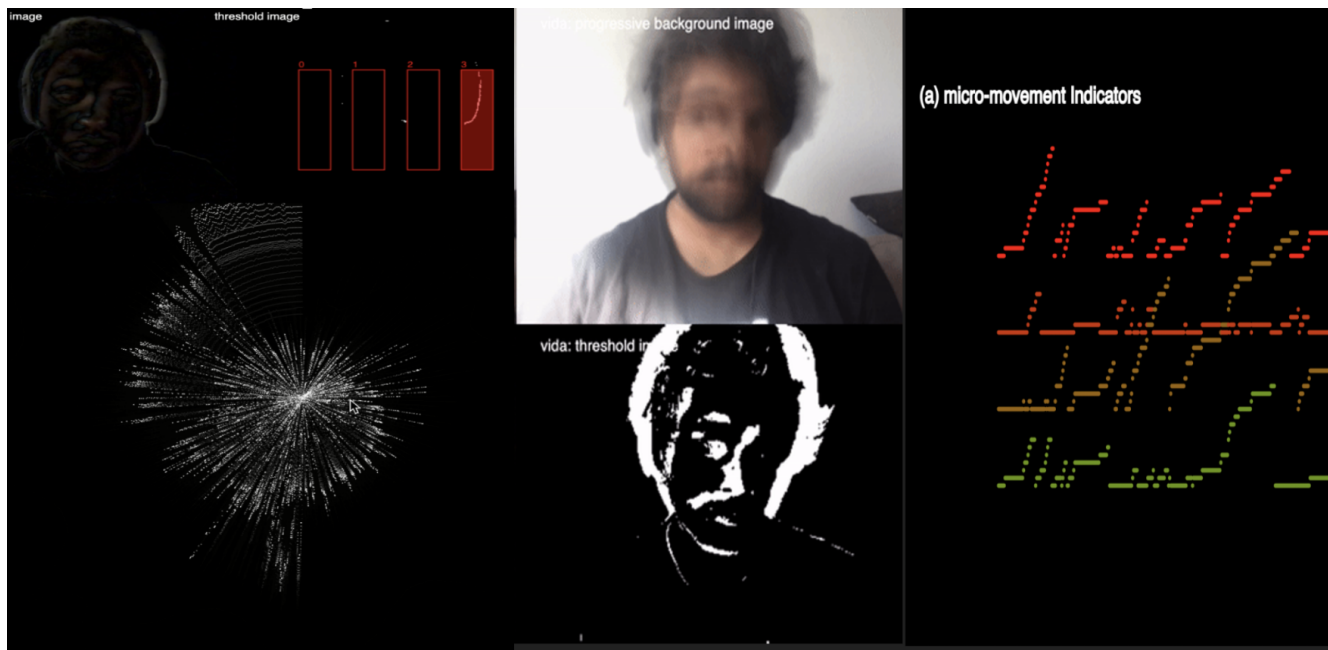
#### 3.1 Flexible Inputs Experiment:

##### **Listen to the Listener:**

In the previous Section we have seen examples of generative music systems that are both - a static sequence of pre-determined instructions (Brian Eno’s Ambient Music) and also interactive algorithms, responsive to inputs from the human performer or collaborator (Bjork’s *Biophilia* mobile game version). Our analysis of the AI Music startups from the recent past shows a trend towards enterprise services that provide generated algorithmic music as a passive background activity for relaxation, games or advertising. On the other hand, generative musical games like those from Brian Eno and Peter Chilvers [83] do allow a listener to actively interact with the composition but through very limited button presses or simple in-game-actions.

In this experiment, we were interested in exploring a dynamic generative music system that can be controlled by an extremely flexible input stream - *movement*. The micro and macro movements of a

listener as they listened to a generative music system were measured and the frequency and intensity of these movements were used to trigger and control different sections of the composition. Micro-movement refers to gentle and small ranges of motions occurring for short periods of time. Macro-movements refers to the larger ranges of motion that in the context of music listening were often periodic indicating - head nodding, head bopping, swinging, singing along and even in some cases dancing.



**Figure 3.1:** Listen to the Listener: Attentive Generative music system that responds to listener movement

### Implementation:

A browser based application was created with the front end developed using *P5js* and a generative composition system in *ToneJS*. Different sections of the composition could be procedurally triggered and controlled based on the macro and micro movement measurements from the video stream. The camera input was used to detect macro and micro-movements as continuously monitored variables. Periodicity detected in the micro and macro-movements would trigger section changes in the composition.

A study was conducted with 15 participants listening to both a static version of the music and our reactive generative music system as alternating control and test groups. A self-reported survey helped identify the moments when the participants were attentive to a change in the music and their corresponding movements indicating attention.

### Analysis:

Camera input and the derived *movement* is a flexible form of input analogous to microphone input and *voice*, described as an input method in the previous section. Such streams of input data as a multi-dimensional continuous variable may contain information, both expected and unexpected by the



system. In the context of this experiment and the accompanying participant study, we identified the different types of micro and macro movements to indicate musical engagement.

The generative music system was unable to handle certain unexpected behavior in the camera input, for example - sudden adjustment of sitting posture, or walking out of the frame. These limitations of the systems were great insights into the capacity of flexible input methods to capture surprising behavior. This experiment was an example of a Generative music system that procedurally composed music with movement as an input to the system.

## 3.2 Slow Creator Experiment

### Artificial FM

One of the main limitations of AI music software, discussed in the previous section, is their severely limited high level controls and inability to produce a wide range of outputs. Most AI music softwares like - JukeDeck, AIVA and Amper Music [41] produce symbolic musical sequences in very few styles (cinematic, folk, pop) with a selection of the key, tempo and length of the music as the only controls.

Jukebox [84] by OpenAI provided a large VQ-VAE (vector quantized variational autoencoder), a generative deep neural network trained on 1.2 million songs crawled from the web (600,000 of which are English). This category of VQ-VAE models have recently (Jan 2021) been used to great success in image generation as well with DALL-E [37]. A unique ability of this model is to generate novel musical continuations from a *priming* segment of audio. The authors demonstrate examples of popular music being used as a prime to generate novel continuations.

#### Implementation:

In our experiment Artificial.fm (July 2021), we created an experimental radio-like platform for generating AI music using the Jukebox model. We collaborated with a few local musicians to submit priming segments to generate new music. (The model and the priming method is described in greater detail in **Section 4.4.1**). The Jukebox model takes about 9 hours to generate 1 minute of audio at 44.1 KHz sampling rates. This is a kind of *slow creator* model which is not ideal for real-time music generation applications.

#### Analysis:

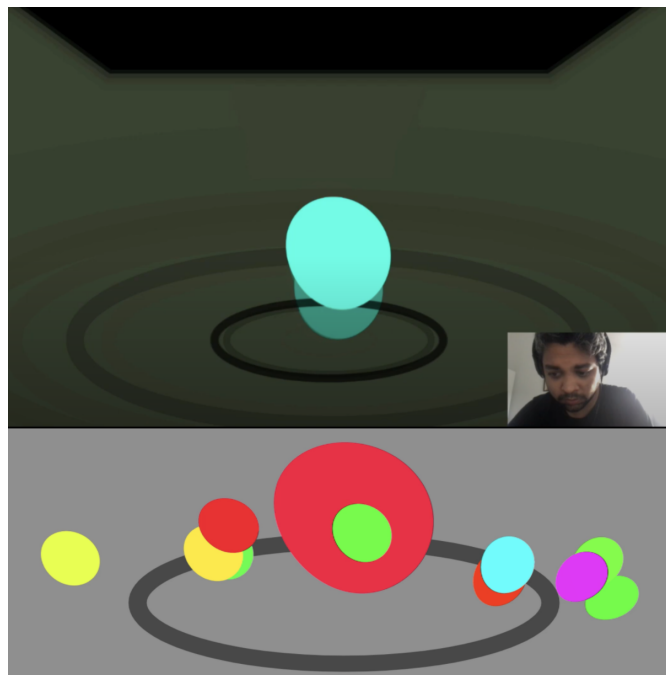
This was a first experiment into generating AI music primed on pre-existing music through a deep learning architecture. Through our experiment we provide a case study for the different stakeholders of this new form of music composition paradigm - the creator of the system and model architectures, the musician whose music was used to prime Artificial.FM streams and the musicians whose music was used in the training datasets used to create the Jukebox model.

In our primary work, we incorporate this idea of a *slow creator* within the context of a brainstorming tool rather than for real-time AI music generation. This experiment showed the limitations of the model to generate long forms of music that are coherently connected with the input audio priming. Instead of using the AI techniques to generate complete pieces of music, we focus on generating short segments of sonic construction material that are strongly connected to the input priming audio. We identify the potential of such a brainstorming tool to provoke serendipity in the composer's ideation stage as a primary design choice in this thesis.

### 3.3 Visual Design experiment

#### Voice controlled Blob Party

In this experiment, we built a server-client architecture for controlling blobs with one's voice through a browser application.



**Figure 3.2:** Design Experiment: Blobs controlled by voice

With the motivation of using Voice as a flexible input medium to interact with AI technologies, we experimented with visual representations of this control. A blob-shaped representation that demonstrated interactive visual behavior to real-time vocal inputs was built.

#### Implementation:

This experiment was done in the pandemic of 2020, as a way to feel connected with a group of peers. A server-client setup built on Javascript was created as a browser application. All participating users would see a central blob shape that they could control surrounded by other blobs of their peers who were active on their respective local computers. The shape and position of the blob would be controlled by the vocal input from the microphone recorded in real time. This was a simple control system that mapped volume and pitch of the vocal input to the blob's visual appearance in the browser. Each blob's shape and movement would be reflected on every user connected to the application. The volume controlled the height of the blob while the pitch controlled the amount of pulsation in its convex-hull representation.

A generative music system built in Javascript played the background music encouraging people to interact with their respective blobs and making them dance in relation to the music. This experiment helped test the design choices of the server-client architecture and the *voice* as an input mechanism represented through the visual language of the blob.

### 3.4 Learnings from Experiments

In this Section, we looked at a selection of initial experiments towards our goal of making an accessible non-programming AI Music composition system using *Voice* as the input medium.

Our experiments with *flexible input* methods to control generative composition systems and moldable visual representations motivated key design choices of our *Living, Singing AI* system. The *slow creator* experiments helped identify various AI algorithms that can be modified to accept *voice* as an input medium. It also motivated our goal of making accessible entry points to engaging in generative AI music technologies. In the previous section we identified the many limitations of current AI tools in a creative practice, the main ones being - lack of control, accessibility and personalization. Through our initial experiments we identify a methodology of using a non-programming interface for active and reflective engagement with generated AI sonic material from a range of models using *voice*.

## Chapter 4

# Living, Singing AI

*"I imagine that as contemporary music goes on changing in the way that I'm changing it what will be done is to more and more completely liberate sounds from abstract ideas about them and more and more exactly to let them be physically uniquely themselves."*

- John Cage  
(Classic Essays on Twentieth-Century Music, 1952)

### 4.1 Living, Singing AI - Philosophy

The *Living, Singing AI* system provides an *intelligent, evolving, scalable, bespoke* composition framework which can be used with just one's *voice*.

As the newer mathematical ideas of AI allow for innovative paradigms of composing and being creative, an accessible ecosystem is paramount to accelerating its assimilation into society. We have shown that AI music continues to be deployed in the limited context of a passive music-as-a-service industry without an emphasis on the individual communicating unique ideas and story-telling - one of the primary functions of music in society.

*Living, Singing AI* challenges this trend by focusing on the unique human composer. It makes multiple modern AI architectures and open-sourced models easy to access, interact with and reflect upon. This ensures that direct engagement with AI music models is not restricted to a niche of composers with programming skills. It opens up interaction with the models to a wide range of amateur and skilled creatives from musical cultures around the world.

#### 4.1.1 Primary Function

## *What does it do?*

*Living, Singing AI* is a composition system that allows a composer to use their *voice* as an input to directly generate a wide range of sonic material. This material is generated by querying three different modern AI music models with vocalized input from the composer. (DDSP, Jukebox and Music Transformer: architectures and relation to input explained in detail in **Section 4.5**).

The composer is then free to score and rank the generated sonic material based on personal reflection and aesthetic preferences. These choices are used to guide an iterative generation of further sonic material. This acts as a brainstorming tool for provoking serendipity in a directed way. All the generated material can then be exported as waveform / MIDI audio files for use in other composition workflows.

Additionally, the system can be used to perform short (30-40 second) compositions from this collection of uniquely personal ideas. These compositions are not meant to be complete musical pieces but rather shorter versions that demonstrate how AI generated sonic material can be incorporated within a context. The goal is to add contextuality to the generated sonic material leaving room for re-interpretation.

## *Who is it for?*

This system uses the *voice* as a means of interaction in a non-programming environment deployed as a browser application. Any kind of amateur or skilled composer regardless of musical culture or technical background can use the system. The system makes interacting with AI music models accessible and personalizable to non-programming composers.

### 4.1.2 Design Choices

#### **Communicate with *Voice*:**

Using voice as the medium for communication with the system, ensures an inviting and flexible design space. The input methodology of a tool is the main interface between a user and their desired goal. Common AI and computer music tools require a familiarity of the input types and limits from the user. Exceeding the limits or violating the rules of the input is usually accompanied by an error message. In AI music systems, as an extension of traditional machine learning methods, it is expected to prepare the correct dimensions, data-types and size limitations of tokenized inputs. Our system eliminates this limitation by inviting the user to make any kind of vocalized sound to begin their interaction.

The *voice* is also a very flexible form of input with no restrictions on the information it may contain - from speech and singing to non-vocalized utterances, vocal mimicry, noise etc.

The vocal timbre is also unique to each individual and one of the most common instruments amongst humans.

#### **Focus on *Reflection*:**

Our system follows an iterative design paradigm with the focus on the unique reflections of the user. After generating new sonic material, the user is invited to score and reflect on each generated sound. These parameters are stored and incorporated into generating further customized material from the AI models. Current AI music softwares act as a single pass from input parameters to generated output and do not support an iterative, reflective composition strategy. This iterative and reflective method was found to support the explore and exploit framework of human-AI collaboration as seen in the examples from the *AI Song Contest*.

#### **Living Representation of ideas:**

A blob shaped abstraction is chosen as the form to represent a user's personalized collection of ideas. This form is chosen as the simplest geometric embodiment of the growing, evolving and moldable nature of the intelligent system.

While some AI systems (Auxuman [85]) embrace the uncanny valley of humanoid representations, we choose to focus on the human composer as the source of creativity and the one being assisted by an intelligent system. The moving and pulsing blob representation of the system is useful to represent the different active and passive behaviors of the system and also invites the user to curiously interact with immediate visual feedback.

#### **Egalitarian System:**

Our system allows the user to interact with and generate sonic material from three different AI models with the same vocalized inputs. It allows the user to reflect on the sonic material produced and its relation to their creative goals without focussing on the method that produced it. Such a system could act as a great framework for an equitable evaluation of different AI models and architectures focussed on the composer's goals.

### 4.1.3 Dimensions of Novelty

In Section 2, we observed the potential of AI as a creative tool through several exemplar paradigms for the AI visual arts ecosystem. We also identified the key limitations of AI music technologies that have so far prevented its assimilation into the wider music making community.

Here, we identify some of the key dimensions of novelty that our system brings to the field of AI musical systems:

**Flexible Input:** In order to allow for surprising uses (and misuses) of the AI models, we propose to incorporate flexible input methods in our systems. We use the *voice* as an accessible and flexible form of *input* to the system. This allows a wide range of input possibilities - speech, singing, noises, vocal mimicry, non-vocalized utterances etc. This sometimes leads to surprising and creative results not intended by the AI models. For example: starting with silence as an input to generate material.

**Slow Creator:** While some algorithms generate sonic material in the order of seconds, others take several hours. We allow for both kinds of models in a slow creator paradigm [86]. Some of the slow generated sonic materials are added to the system as they become available and can be evaluated with equal value as those generated from faster algorithms.

**Adapts to User:** We present a bespoke system where all sonic material generated is centered around the unique vocalized queries of the individual user. Additionally, the user can shape the system by their preferences over time with a method of selecting and scoring the generated content. AI Musical instruments present this unique ability to adapt and evolve their functionality and form with each individual user.

**Composer Focused Evaluation:** Our system focuses on the sonic material generated and allows composers to evaluate their aesthetic and creative goals regardless of the techniques they are generated from. We demonstrate this with a selection of three AI models but this could be extended to incorporate newer AI music models as they are deployed. This will allow creators of new AI music models to evaluate their methods in comparison with existing models, with a focus on the composers and how they use the tool for their personal creative goals.

## 4.2 Definitions

In this section we describe the key definitions of concepts in our system followed by a detailed description of each sub system.

### *Intelligent:*

Our system not only uses multiple artificial intelligence models to generate derivative sonic material from initial vocalized ideas by the human user, it also allows the individual to shape the generated material by a method of scoring and selecting. This allows for a context aware generative system that iteratively grows in accordance with the user's choices. Different behavior modes can be set to make the outputs produced less or more exploratory. The classic textbook *Artificial intelligence: a modern approach* [87] defines intelligent agents (IA) as "anything that can be viewed as perceiving its environment through sensors and acting upon that environment"

## Evolving:

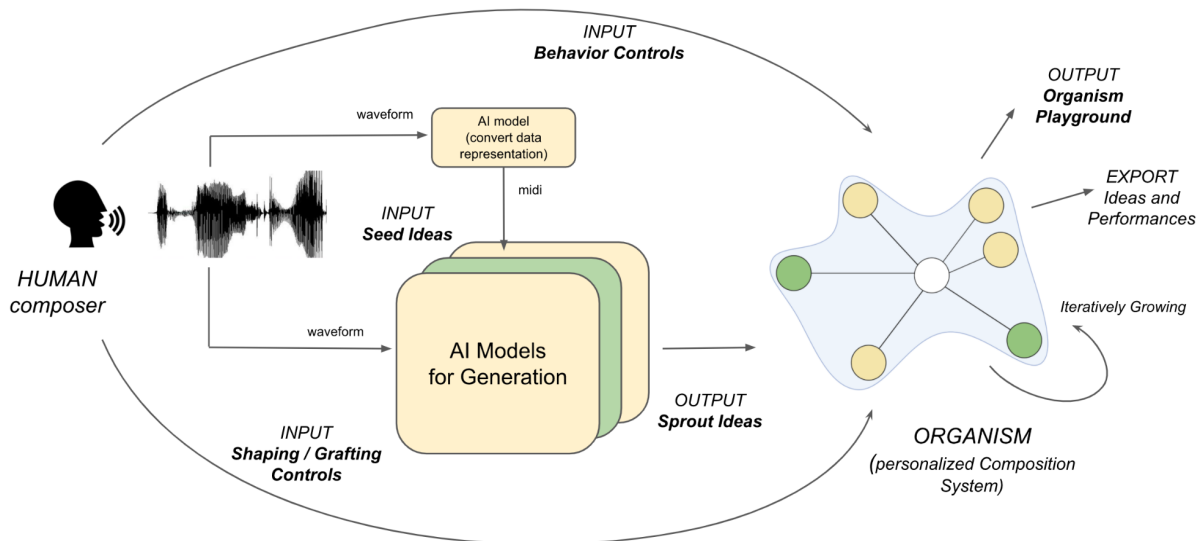
Our system can be uniquely shaped and grafted by the user. These unique preferences are used to iteratively generate further sonic material. This demonstrates an evolving characteristic of the system.

## Bespoke

The system is designed to be generative with the individual user at its center. It begins with a real-time capture of vocalized ideas from the microphone that are unique to the individual at the time they were produced and allows further shaping of the outputs. This creates a bespoke system that is uniquely personalized for each user.

## Scalable

*Living, Singing AI* runs in a client-server design where the client is a browser application consisting of the user interface. The server is a web application that wraps the underlying software packages providing an API for its functionality. This presents an ecosystem which can be expanded to multiple users with access to a browser and the internet.



**Figure 4.1** : Overview of the *Living, Singing AI* system

**Seed idea:** The vocalized query recorded using the microphone from the user is called the *Seed idea*. This idea is used as the starting point for all the AI models and is therefore identified as the Seed of the system.



**Sprout idea:** The Seed idea is then used as an input in multiple generative AI models to create a collection of derivative sounds. Each unit of generated sound, sample and MIDI sequence from these models is called a *Sprout idea*.

**Organism:** This collection of Sprout ideas is called an *Organism*. The Organism is personal to each individual because it is generated from their unique *Seed ideas* and its generated *Sprout ideas*. The Organism can also be shaped by other inputs from the user that personalize the Organism.

**Vocabulary:** The Organism's *vocabulary* is the current list of Sprout ideas stored in the database for the unique Organism.

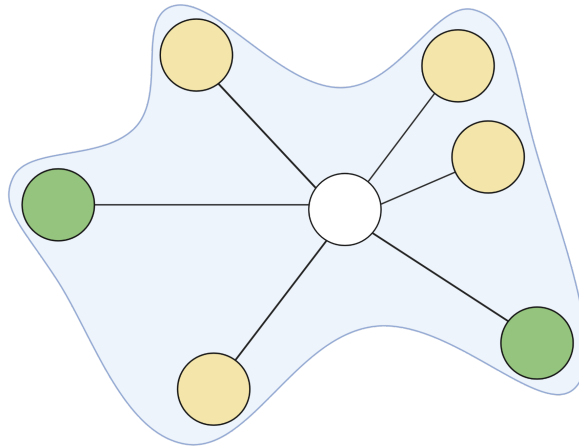
**Behavior:** The Organism has different behavior modes that can also be controlled as inputs from the user

**Grafting:** The Organism's individual Sprout ideas can be individually deleted, annotated, scored and ever used as further Seed ideas for the Organism. The act of interacting with the Organism via these controls is called *Grafting*.

## 4.3 System Overview

*Living, Singing AI* enables a user to interact with a browser application to record and send audio or a 'Seed Idea' that gets sent to a database. This audio is then used as an input to three different generative artificial intelligence model APIs (DDSP, Jukebox, Music Transformer, explained in **Section 4.5**). This produces multiple audio files or *Sprout ideas* that are sent back to the browser application. Once the *Sprout ideas* have been generated, the user can interact with the generated results in two different modes - a *Grafting Control* and a *Behavior Control*. This results in personalizing the system of *Sprout* and *Seed* ideas, collectively your *Organism*.

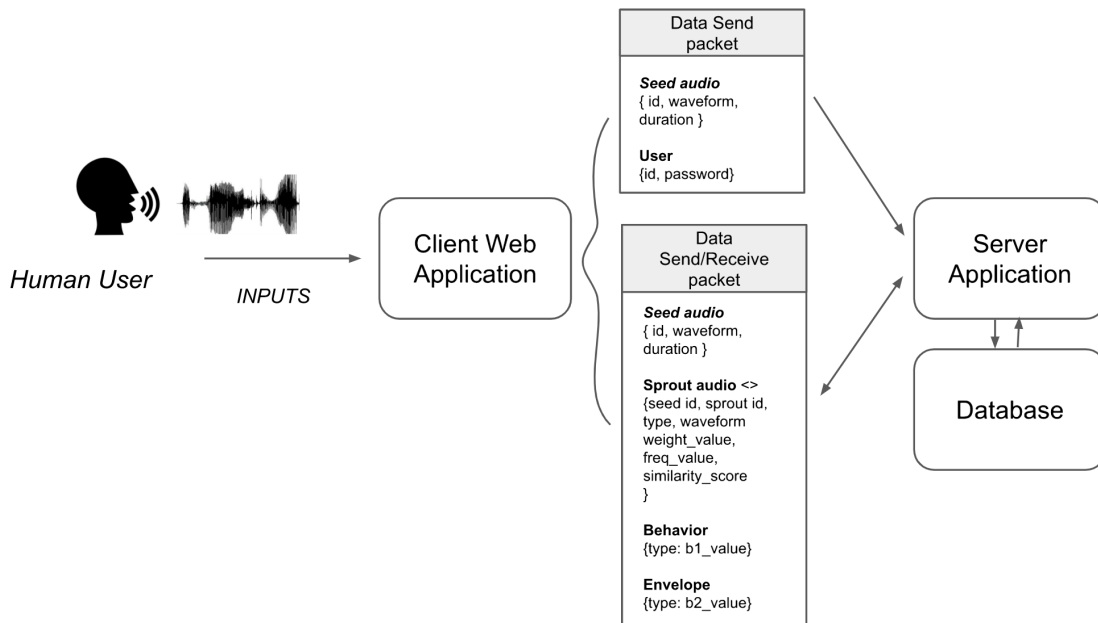
The following section describes the implementation of each part of the system architecture followed by a walkthrough of the interactions in a sequential manner through the three sections - "Create your Organism", "Perform with your Organism" and "Proliferation of your Organism".



**Figure 4.2 :** Visual representation of an example '*Organism*'. White circle represents - '*Seed idea*', green and yellow circles represent different kinds of '*Sprout ideas*' from the different generative models. This *Organism* has a shape and collection of generated sounds unique to the user that created it.

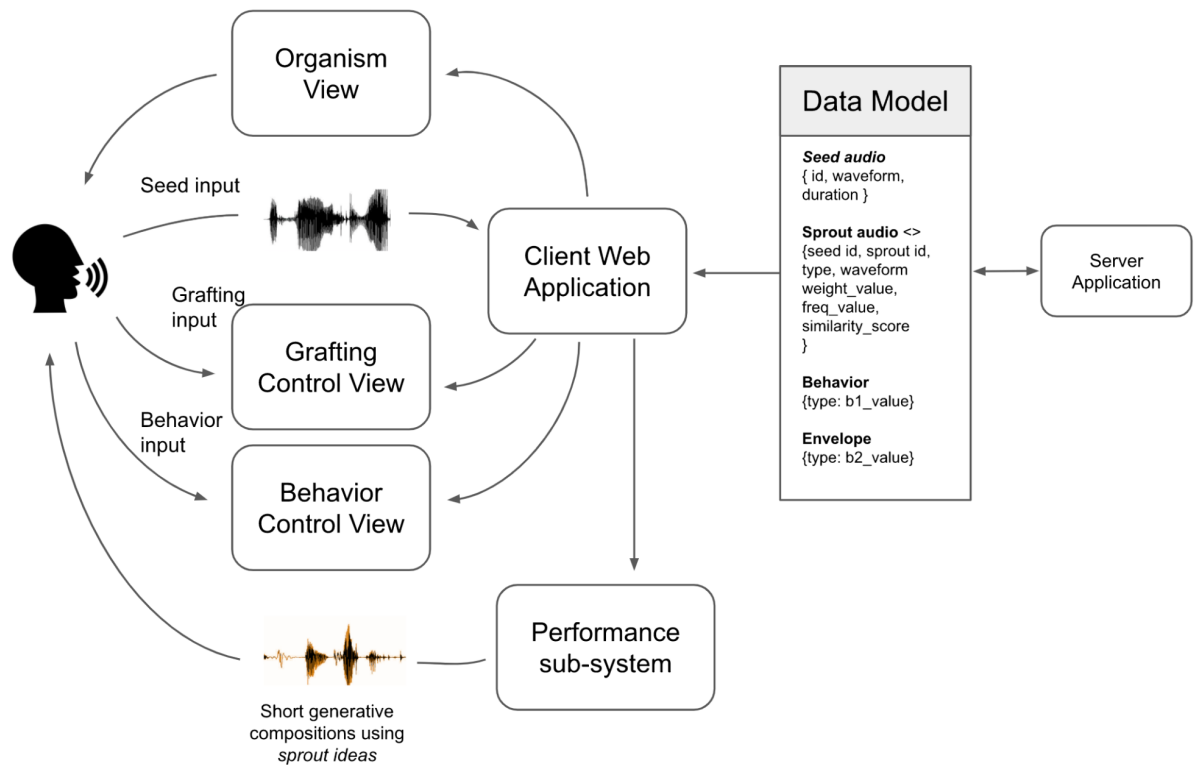
### 4.3.1 Web Application

The *Living, Singing AI* system is implemented in Python3 and Javascript. The front end is written in Javascript using ReactJS and the user interface was prototyped on a local server using P5js. The human user interacts with this front end through which they can record and send their *vocalized queries* and interact with the system. The backend is written as a Flask server that provides a REST API to the functionality. It implements a number of API endpoints to send and receive metadata in the form of a json query. This overall structure is described in Figure 4.3.



**Figure 4.3:** Web application client - server model

## A. Client:



**Figure 4.4:** Client side system flow diagram

The client web application handles the following different components. Each of the input interactions are explained in detail in **Section 4.3.3: User Interface**.

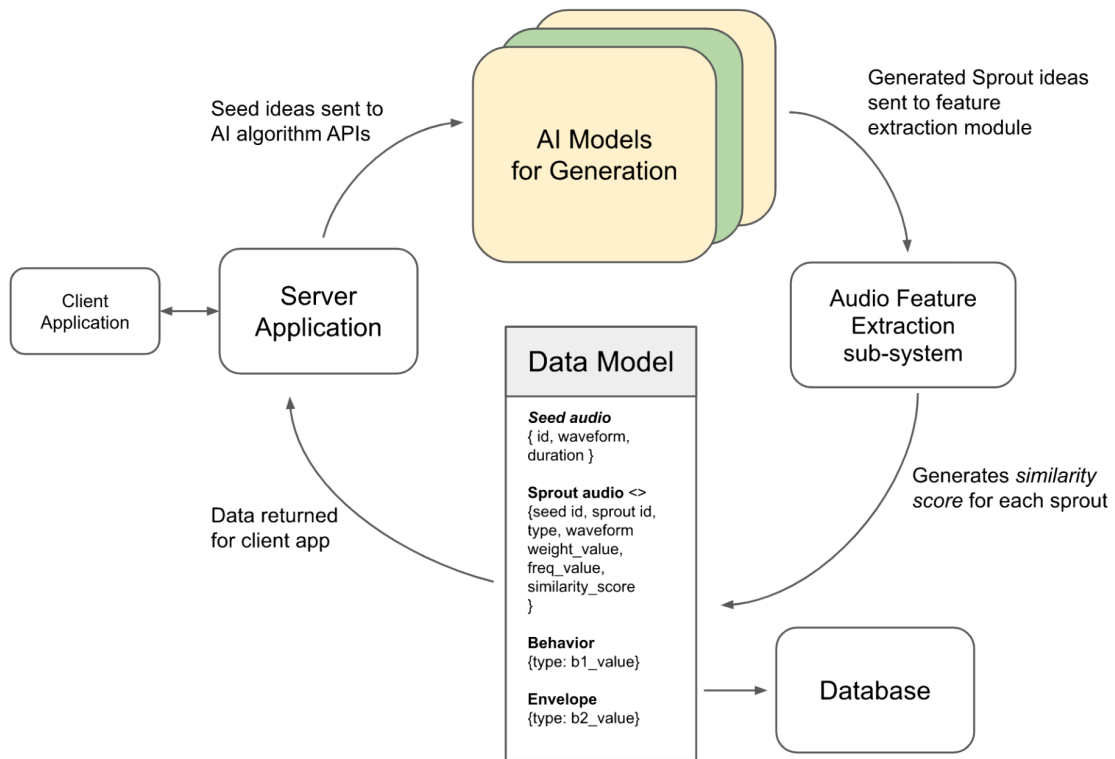
*Organism View:* Handles the visual representation of the Organism including its internal Seed and Sprout ideas. Clicking on any of the Seeds or Sprouts plays back the audio corresponding to it.

*Grafting Control View:* Handles the controls for shaping and grafting the Sprout ideas for your Organism. It shows the different options for shaping your Organism that the user can interact with.

*Behavior Control View:* Handles the controls for the behavior of the Organism. Interacting with this view allows you to perform with your generated sounds.

*Performance Sub-System:* Handles the creation of short generative compositions depending on the selected behavior controls. Each composition is about 30-40 seconds long.

## B. Server:



**Figure 4.5:** Server side system flow diagram

The server is implemented as a Flask application and interfaces with the three generative AI algorithms - Jukebox, DDSP and Music Transformer. The underlying AI Models are explained in detail in **Section 4.4: Underlying Models**. The Audio Feature Extraction sub-system iteratively updates the Sprout data and is explained in detail in **Section 4.7: Proliferation of your Organism**.

### 4.3.2 Data Model

The Organism collectively represents all the ideas generated by the human composer and the generative AI algorithms. This information is represented and stored in a Data Model which consists of four parts - *The User Data* - which uniquely identifies a particular user, *The Seed Data* - the unique vocalized queries from the user, *The Sprout Data* - the generated sonic material from the underlying AI models and *The Performance Data* - the unique grafting and behavior controls for each Organism.

**User data:** {id and password} - Uniquely identifying each individual user

**Seed data:** {id, waveform and duration} - Each Seed shares the user id uniquely identifying it. It also stores the sound file and length of the file.

**Sprout data** : {Seed\_id, Sprout\_id, type, waveform, weight\_value, freq\_value, similarity\_score} -

Seed\_id, Sprout\_id: Uniquely identifying Sprout\_id for each Sprout and the parent Seed id.

type: Represents the type of algorithm that produces the Sprout idea - (1) Jukebox, (2) DDSP or (3) Transformer.

waveform: Represents the audio file corresponding to the generated Sprout idea.

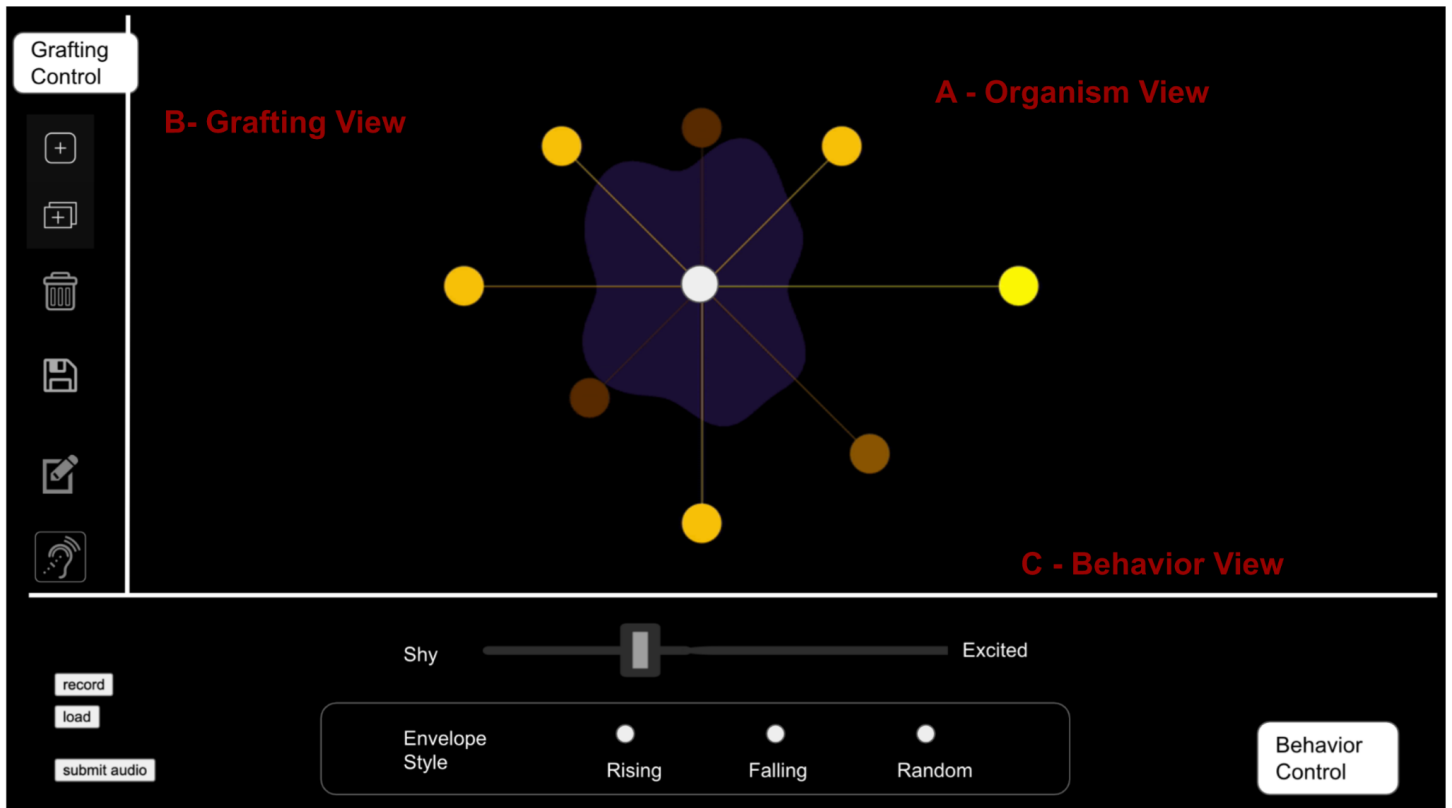
weight\_value, freq\_value: These values are assigned to each Sprout idea by the user through the grafting controls and represent how interesting the user finds the particular Sprout.

similarity\_score: This is calculated as the audio feature similarity between that particular Sprout and the highest weighted Sprout at the audio feature extraction sub-system. Sprouts with the lowest similarity score are iteratively deleted.

**Performance data** : {behavior, envelope} - These values are assigned to the Organism by the behavior controls. They represent the behavior state (0-shy to 5-excited state) and envelope shape (rising, falling or random).

### 4.3.3 User Interface

The front-end presents the three views to the user to interact with their generated Organism. This *User Interface* is chosen to be intuitive and visual to the non-programming musician. Clicking on the “Grafting Control” button on the top-left and the “Behavior Control” button on the bottom-right reveals the corresponding panels seen in Figure 4.6.



**Figure 4.6:** User interface view showing Organism View, Grafting Control View and Behavior Control View.

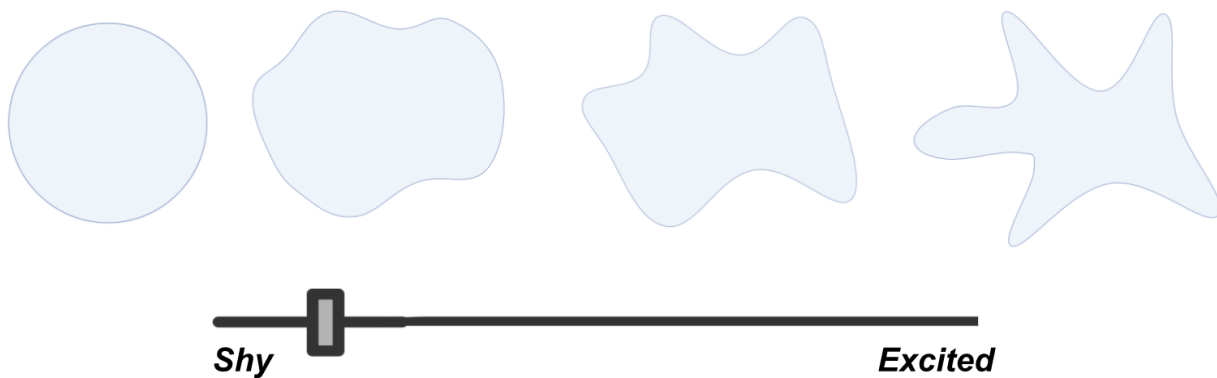
### A. Organism View:

The Organism view is in the center and shows a pulsating and moving blob representation. Clicking on the Organism shows its internal structure of *Seed* and *Sprout* ideas. The central white circle represents the ***Seed idea***. The other surrounding circles represent the ***Sprout ideas***. Clicking on each Seed or Sprout plays back the corresponding audio.

#### *Blob Representation:*

This abstraction of a blob shape was chosen to represent one's *Organism* and the collection of a user's personalized ideas. This simple geometric shape constantly pulses and changes in shape indicating a sense of an organic living form. The simplest living forms in nature - whether single celled like bacteria or multicellular like amoebas are composed of a cell wall that contains, within it, all the parts that constitute it. Our Organism and its constituent Seed and Sprouts are also similarly contained within the blob shape. This blob abstraction suggests the basic characteristics of our system - Intelligent, Evolving, Growing and Moldable.

The blob shape also conveys the different behavior modes that are used in performing with the Organism. A simpler shape approaching a circular convex hull indicates a less active “Shy” mode and a more complicated concave shape as shown in Figure indicates a more active “Excited” mode.



**Figure 4.7:** Blob shape representing Shy to Excited scale of behavior modes

#### *Internal Structure Representation:*

The line and circle - tree-like internal structure representation appears when you click on the Organism. This representation was chosen to indicate the Sprouts and Seed relation where the Sprouts are directly related to the Seed that generated them. The Sprouts, regardless of which generative algorithm produced them, appear equally distributed in the internal structure representation.

All the Sprouts begin at an equal distance from the Seed in the center and are then given different radii and brightness, to represent the ways they are *Grafted* by the user. This *Grafting Control* is visually represented through its proximity to the central Seed. A Sprout that is farther away from the Seed has been judged to be *less interesting* by the user and a Sprout and a Sprout that is nearer is judged to be *more interesting*.

#### **B. Grafting Control View:**

The Grafting Control view is on the left side panel. It contains the different grafting options in a vertical menu format and allows you to interact and mold your *Organism*. Each of these controls is explained below in **Section 4.5.2: Grafting Controls**.

#### **C. Behavior Control View:**

The Behavior Control view is on the bottom panel. It has a horizontal view representing the Behavior controls - a slider for the behavior mode and an option to choose an Envelope style. Each of these controls is explained below in **Section 4.6.2: Behavior Controls**.

## 4.4 Underlying Models

## 4.4.1 AI Models for Generation

Modeling sequences is an important sub-problem of Artificial Intelligence under which music as a sequence of symbols / samples presents unique challenges. This is because of the following two reasons:

- (a) The semantic hierarchies of musical information are complex, subjective and work over multiple scales of time (in the order of millisecond timings to several minutes for a motif or phrase);
- (b) Audio representations at the sampling rates of human hearing make the sequences to be modelled really long, i.e. a typical 4-minute song at CD quality (44 kHz, 16-bit) has over 10 million timesteps.

Transformer models [88] introduced in 2017, outperformed (in the context of natural language modeling) the previously popular sequence modeling methods of Long Short Term Memory (LSTM) neural networks (an improvement on the basic Recurrent Neural Networks (RNN)) by introducing a novel architecture of just attention models. This attention mechanism looks at an input sequence and decides which other parts of the sequence are important. Prior to this, many RNN models had been applied to music like ImprovRNN, melodyRNN [89] etc. These modelled a MIDI sequence as a language with limited success. Over the next few years, the success of transformer models became apparent when really large language models (GPT-3 with 175 billion parameters introduced in 2020), demonstrated exceptional generative abilities [90].

Jukebox from OpenAI, April 2020 [84] and Music Transformer from Google, Dec 2018 [91] are the two largest Transformer based models for music currently available. While Jukebox generates raw audio, Music Transformer generates symbolic piano performances as MIDI sequences.

In addition to these, Differential Digital Signal Processing (DDSP) [92] architectures were introduced by Google Magenta in Jan 2020, with a unique approach to modeling musical neural networks. Unlike typical blackbox end to end deep learning, DDSP breaks down the modeling of music into traditional signal processing blocks. Training individual models for each block, allows for a modular and interpretable architecture.

These landmark models were chosen as the three generative AI models that we include in our system to generate *Sprout ideas* from a *Seed* vocal query by the user.

### **Jukebox Model:**

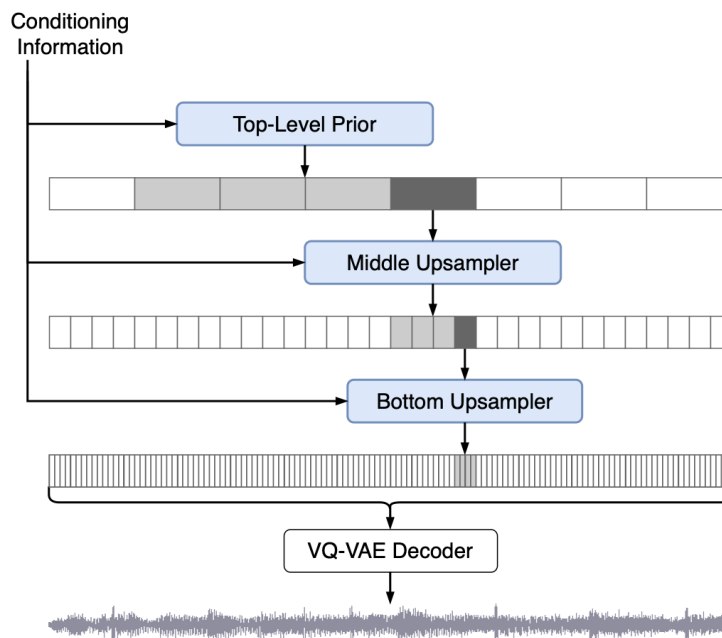
The Jukebox Model from OpenAI [84] is trained on 1.2 million songs conditioned on lyrics, genre and artist names. This database of songs has been scraped from the web and includes 600,000 English songs.



It captures raw audio waveforms at three different hierarchical levels of downsampled resolutions in the form of VQ-VAE codebooks (vector quantized variational autoencoders). It is also able to generate audio conditioned on a ‘priming sample’ by continuing the sequence of codebook vectors at each hierarchical level. It first generates the VQ-VAE codes for the priming audio and then uses a conditioning on the codes to generate future samples from the coarse downsampled top level to the fine upsampled bottom level.

Figure 4.8 shows this hierarchical windowed sampling, indicating the three levels of the codebook encoding. At the top level the audio data is compressed by 128 times and divided into longer segments where the codebook captures coarse representations of audio, but with long term dependencies  $\sim 20$  seconds. At the bottom level the audio data is compressed by 8 times and the corresponding codebooks capture finer details  $\sim 1.5$  seconds. Through this hierarchical nature combined with vector quantized codebooks, the model is able to capture really long term dependencies previously not possible.

This primed sampling is the technique we use in our system with our Seed audio as the prime. Since this is a non-deterministic method, we generate, in parallel, multiple versions of the continuation simultaneously. This enables us to create a collection of *Sprout ideas* from a single *Seed idea*. Each output from the same primed audio can be widely different while still capturing audio information (timbre, rhythm, melody) from the primed samples. This is a difficult model to control because of the very limited input parameters; it is also extremely slow to run. (9 hours to fully render 1 minute of upsampled audio).



**Figure 4.8:** OpenAI Jukebox hierarchical VQ-VAE structure.

Here we show an example from our system to elucidate:

[Sample Seed Idea] =====> [Sample 1 Sprout Idea generated by Jukebox]  
 =====> [Sample 2 Sprout Idea generated by Jukebox]

Despite its limitations, *Sprout ideas* generated from this method were found to be the most surprising, interesting and varied. Since the vector codebook look up step is non deterministic, multiple runs of the same priming *Seed* audio generate a slightly different sequence of codebook vectors. Every windowed set of codebooks is conditioned on the previous vectors and so over a few time steps it generates a completely different codebook vector sequence which in turn results in completely different audio. The generated audio continues to maintain coherence through each hierarchical step.

### DDSP Model:

The Differential Digital Signal Processing model [92] is trained on 10 minute audio recordings of monophonic instruments - Violin, Flute, Saxophone and Trumpet. It is a modular system of autoencoders for each component of a sinusoidal modeling process. Typical sinusoidal modeling involves modeling audio as having a sinusoidal component and a stochastic noise component, which in this case is modeled as separate autoencoders.

The harmonic audio, filtered noise and reverb modules conditioned with fundamental pitch (F0) are individually trained for each instrument dataset. This allows for any new audio to be fitted with the harmonic and noise component of a different instrument's pre-trained autoencoders.

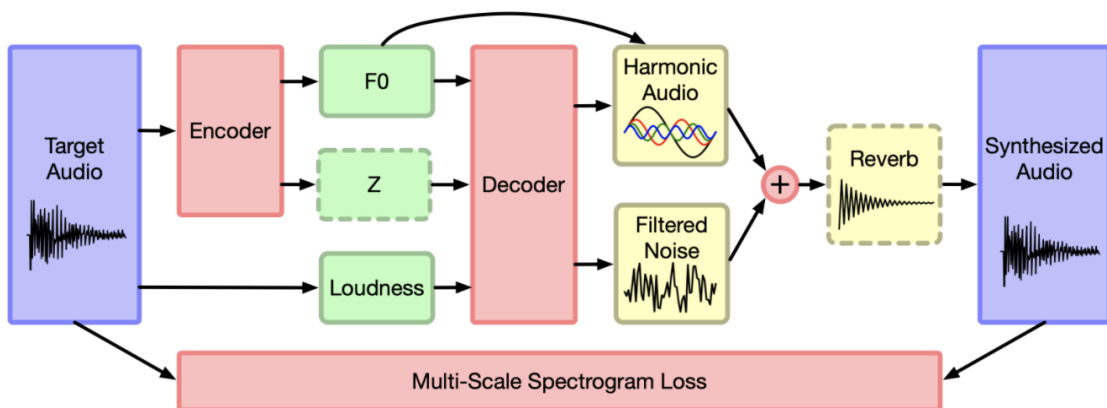


Figure 4.9: Google Magenta's Differential Digital Signal Processing architecture [92]

In our system we use the *Seed audio* as an input and run it through the Violin, Flute, Saxophone and Trumpet timbre models that have been openly made available. This creates *Sprout ideas* that follow the fundamental pitch of the Seed audio but have the characteristics of another instrument. Since it is a waveform based method, the noise and reverb components of the DDSP are able to capture unique properties (like breathing noise for Flute playing) of the instruments that previous models haven't been able to capture.

Here we show an example of the same *Seed idea* converted into different *DDSP Sprout ideas*.

[\[Sample Seed Idea\]](#) =====> [\[Sample 1 Sprout Idea generated by DDSP\]](#)  
=====> [\[Sample 2 Sprout Idea generated by DDSP\]](#)

### Music Transformer:

The Music Transformer [91] is an attention based neural network that captures long term coherence from symbolic piano playing performances. It is trained on the datasets - JSB Chorales (382 Chorales from J.S.Bach) and Piano-e-Competition datasets of ~ 200 hours of human piano performances. It is able to produce continuations and melody conditioned accompaniments of a given motif while maintaining a limited degree of temporal coherence. In our system we convert our *Seed audio* into MIDI using Google Magenta's Onsets and Frames model [93] and then pass it as a priming sequence to the Music Transformer model. The generated MIDI continuations and accompaniments are returned as the *Sprout ideas*.

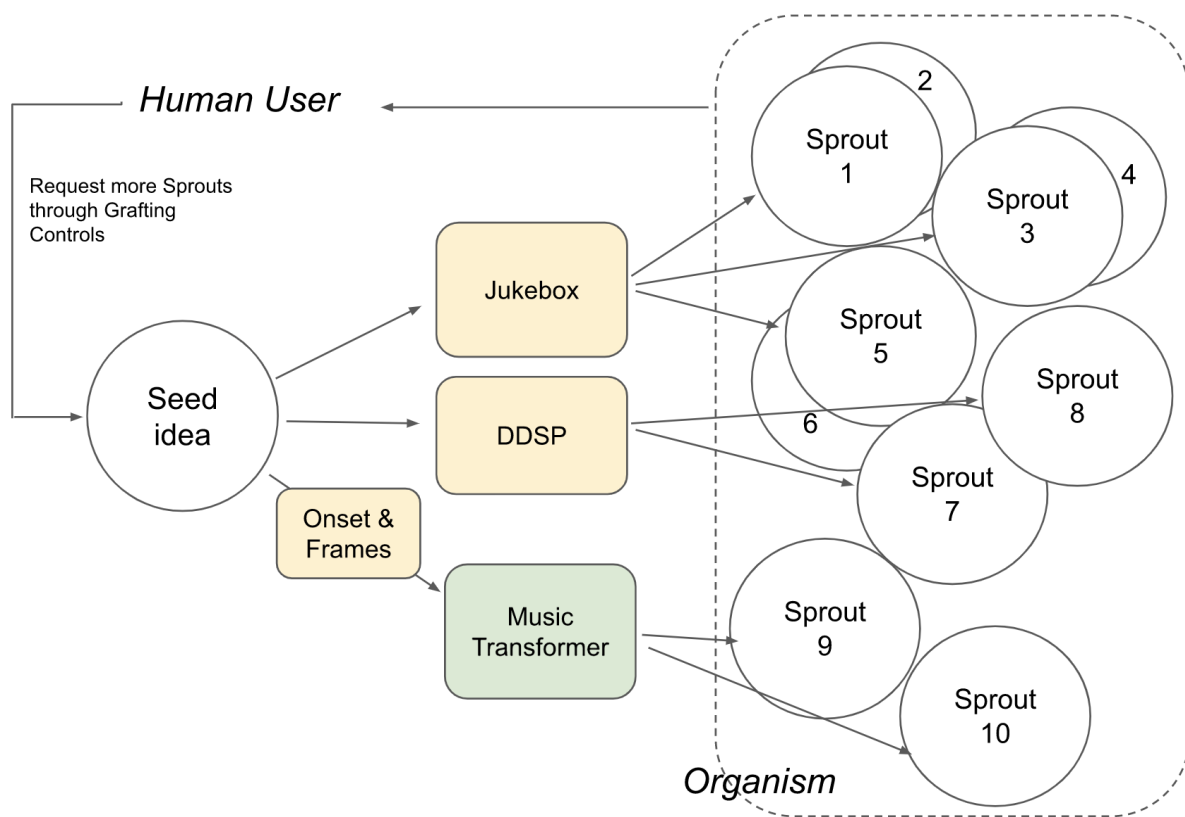
Here we show an example of a MIDI piano generated from the same Seed idea using Music Transformer.

[\[Sample Seed Idea\]](#) =====> [\[Sample 1 Sprout idea generated by Music Transformer\]](#)

## 4.4.2 Generation Method

Everytime a *Seed idea* is created and submitted to the server, the three generative AI algorithms are run creating a collection of Sprouts. Jukebox generates multiple continuations in parallel, so we generate 6 Sprouts at a time and for the remaining two algorithms DDSP and Music Transformer, we generate 2 Sprouts by calling the APIs twice. This results in 10 Sprouts generated at a time.

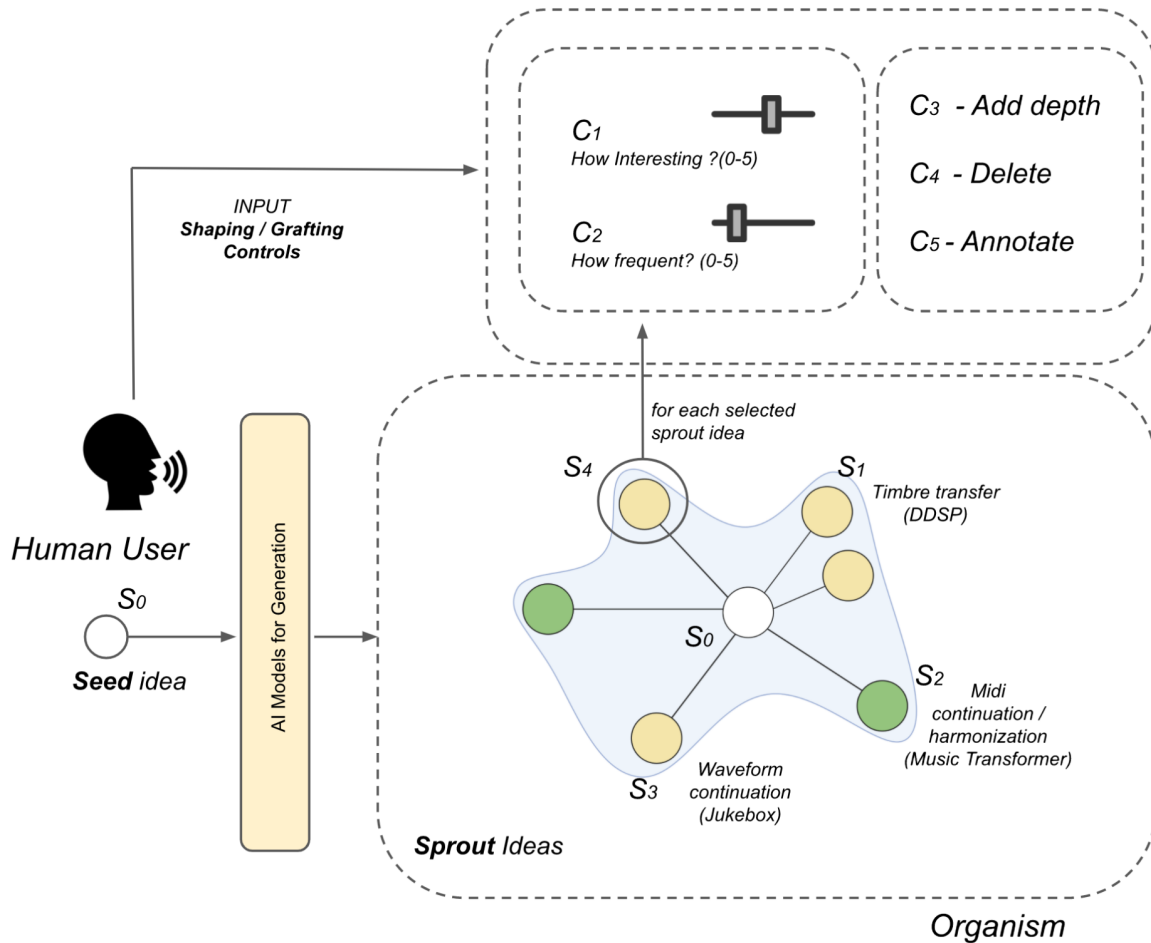
Further Sprouts can be generated by the user through the grafting controls explained in the next section.



**Figure 4.10:** Generation Schematic from Seed to Sprouts

## 4.5 Creating *your* Organism

On starting the system, the user interface prompts you with a record button to create a vocalized query at the browser via the microphone. This creates your *Seed idea* that gets submitted to the server. The server returns the results from the different AI models (Figure 4.10) for generation in the form of your *Organism*.



**Figure 4.11** Schematic Diagram of Creating your Organism by input Seed idea and Grafting controls.

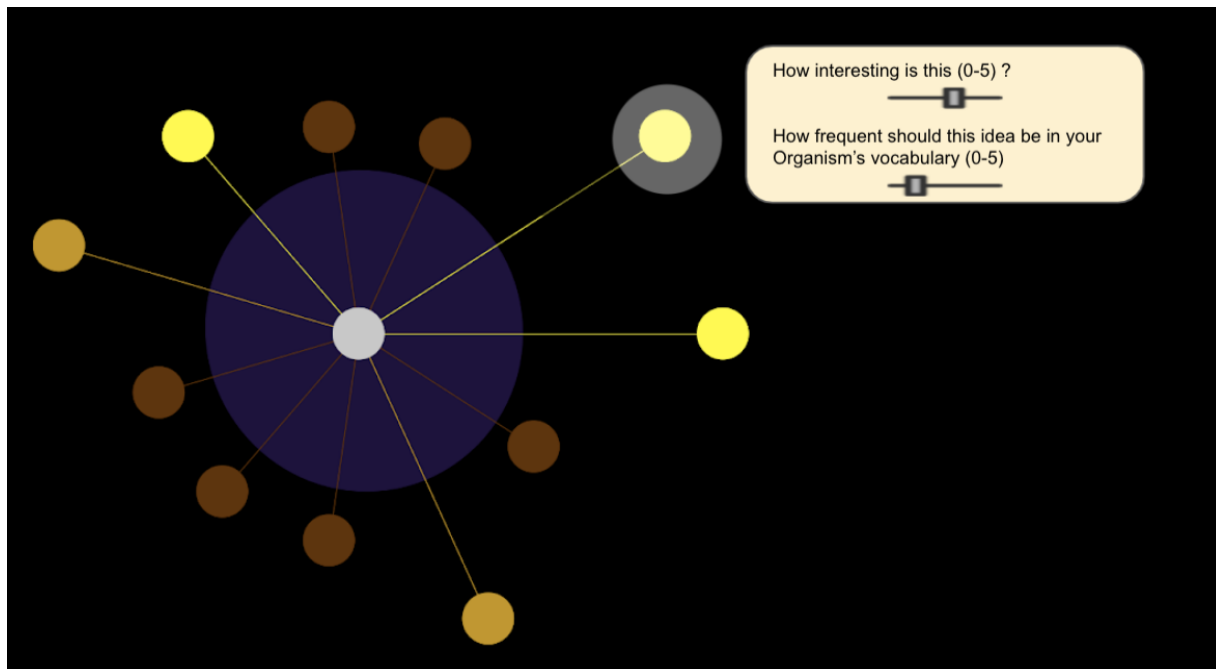
Figure 4.11 describes the schematic of the Organism structure at creation.  $S_0$  is the Seed idea submitted by the user.  $S_1$  is generated from the DDSP model,  $S_2$  is generated from the Music Transformer model and  $S_3$  is generated from the Jukebox model.  $S_4$  is an example of a selected *Sprout* for which we have the grafting controls -  $C_1$  through  $C_5$ .

### 4.5.1 Vocalized Control

The direct way to interact with the system is by submitting a vocalized query. This is stored as the **Seed idea**. All generated **Sprout ideas** are directly related to the uniquely submitted Seed. It is the first form of interaction with the system. It allows the user to query the three underlying AI models directly with just the voice as a control.

## 4.5.2 Grafting Control

The next way to interact with the system is through the *Grafting Controls*. Once a collection of *Sprout ideas* are returned in response to the original *Seed idea*, the user can reflect on the outputs generated. This is done by allowing the user the following controls for each selected *Sprout*:



**Figure 4.12:** A selected Sprout and its accompanying Grafting Controls (C1 and C2).

### **C1** - How interesting is this (0-5) ?

This lets the user reflect on the generated Sprout and give a score to the Organism. A higher score is represented with the Sprout appearing closer to the center Seed, and a lower score pushes it farther away. This score is stored in the Data Model as *{weight\_value}* of the Sprout. This is used in the Proliferation of the Organism as new material is generated.

### **C2** - How frequent should this idea be in your Organism's vocabulary (0-5) ?

This lets the user decide if the generated Sprout should be rare or frequent when the Organism is in performance mode generating short compositions. This is stored in the Data Model as *{freq\_value}*. It is possible that a very interesting Sprout is chosen to be frequent while another interesting Sprout is chosen to be rare.

**C3**  **Add depth**

This allows the selected Sprout to be submitted to the server as a new *Seed idea* as a source input to generate further *Sprout ideas* from. This creates an iteratively growing collection of sounds.

**C4**  **Delete**

This allows the selected *Sprout idea* to be deleted from the Organism.

**C5**  **Annotate**

This allows the user to give a name to the selected *Sprout idea*.

**C6**  **Export**

This allows you to export all the generated *Sprout ideas* as wave and MIDI files respectively to save to a local drive

**C7**  **Listen**

This allows you to listen to short generated compositions (30-40 secs) using the generated Sprout ideas and the Behavior controls set for your Organism, described further in Section 4.6.

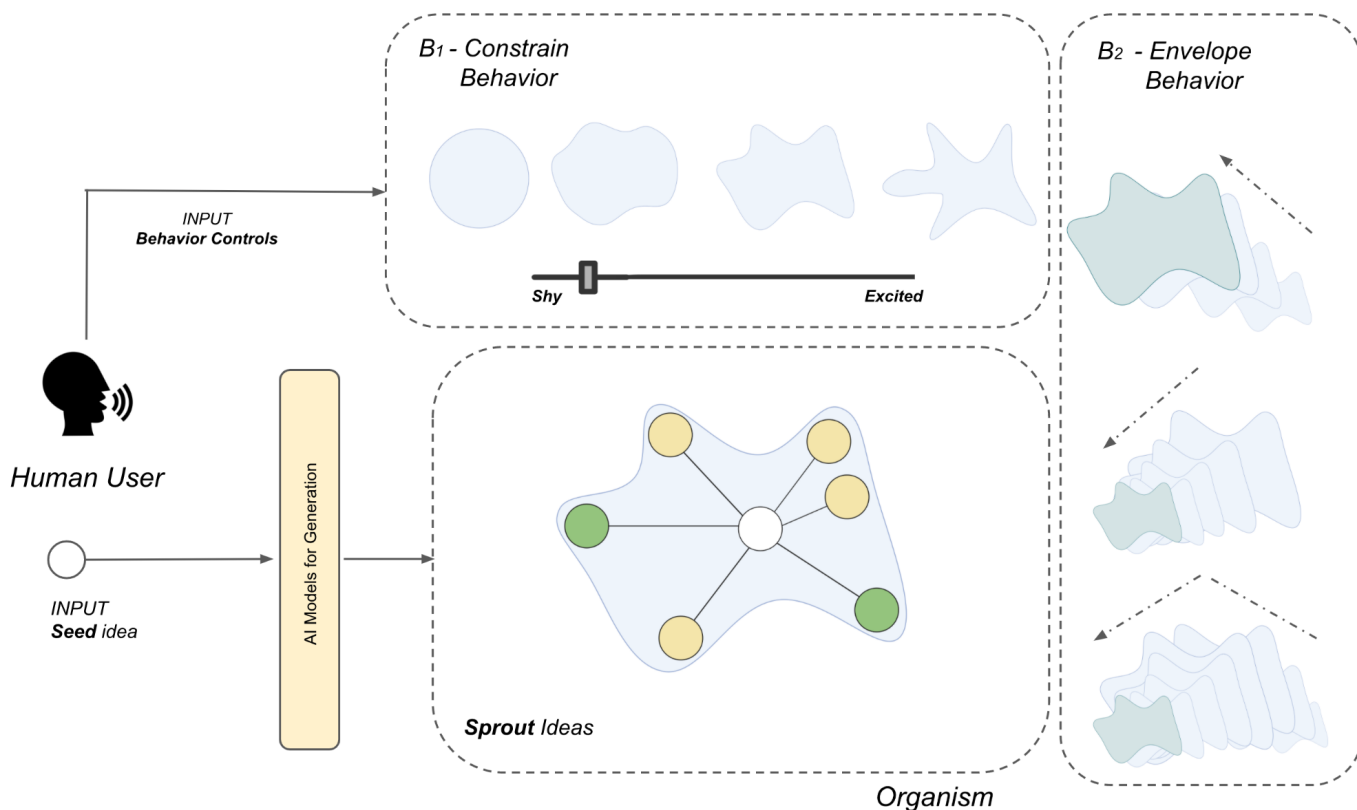
## 4.6 Performing *with* your Organism

### 4.6.1 Objective

After creating and *grafting* your *Organism*, the Organism can be given certain behavior properties to generate short mutable compositions using the Sprouts. Incorporating the generated sonic material into short compositions is intended to add contextuality for provocation. The individual Sprout ideas are provided further context when played back in these compositions leaving room for re-interpretation by

the composer. This also adds a performability to the generated Organism which can be controlled by the *Behavior control* inputs.

The *behavior controls* affect two different properties of your Organism - the choice of Sprouts for the generated compositions and the sequence of the chosen Sprouts. When the Organism is in its performance mode, accessed by pressing either the 'Listen' button or when the '*Behavior Control*' panel is visible, it produces short phrases of generative compositions by stitching together the *Sprout ideas*. These compositions are shaped by the following behavior controls:



**Figure 4.13:** Schematic Diagram of Performing with your Organism with Behavior Controls

## 4.6.2 Behavior Controls

### **B1 - Constrain Behavior**

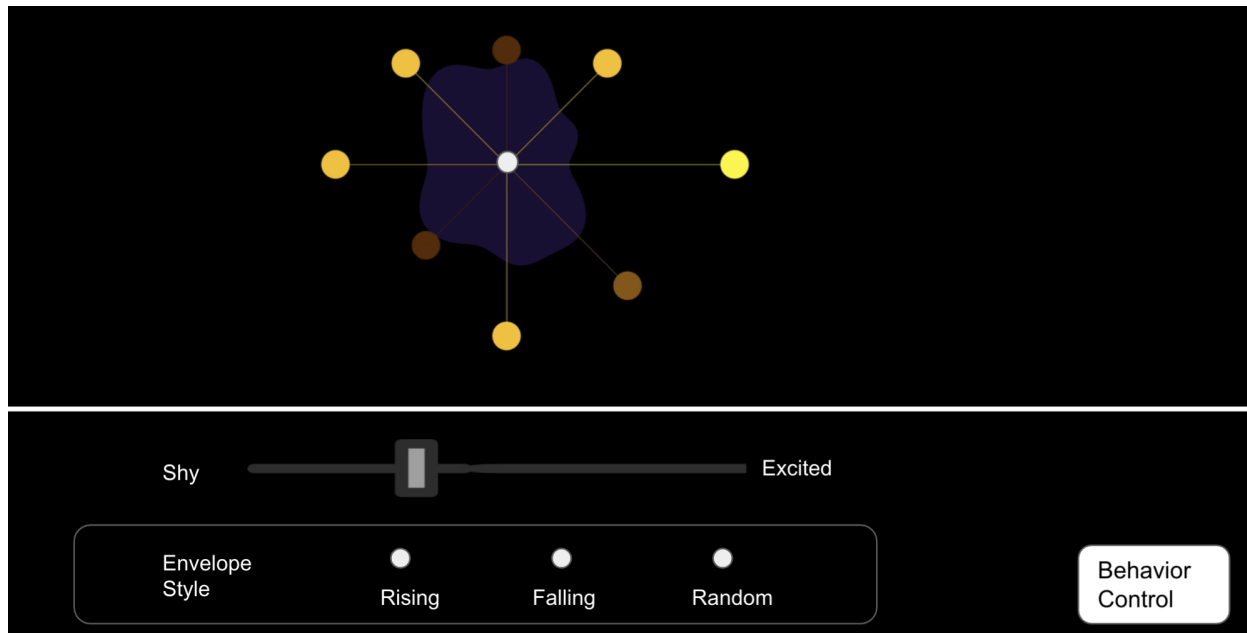
This is a value (0-5) that is a slider where 0 represents '*Shy*' behavior and 5 - '*Excited*' behavior. In the shy mode, the Organism produces sounds from the *Sprout ideas*, but chooses only the higher weighted (higher rated on the interesting scale) sounds. It also generates sparse compositions with more silences.

### **B1 - Envelope Behavior**



This is a selection between three kinds of envelope shapes - (a) Rising, (b) Falling or (c) Random. This determines the shape of the generated composition.

A sequence of *Sprout ideas* are chosen with increasing, decreasing or randomly ordered *complexity* depending on the chosen Envelope style. For each *Sprout idea* its complexity is calculated by the Performance sub-system.



**Figure 4.14:** User Interface view showing Behavior Controls in Performance mode.

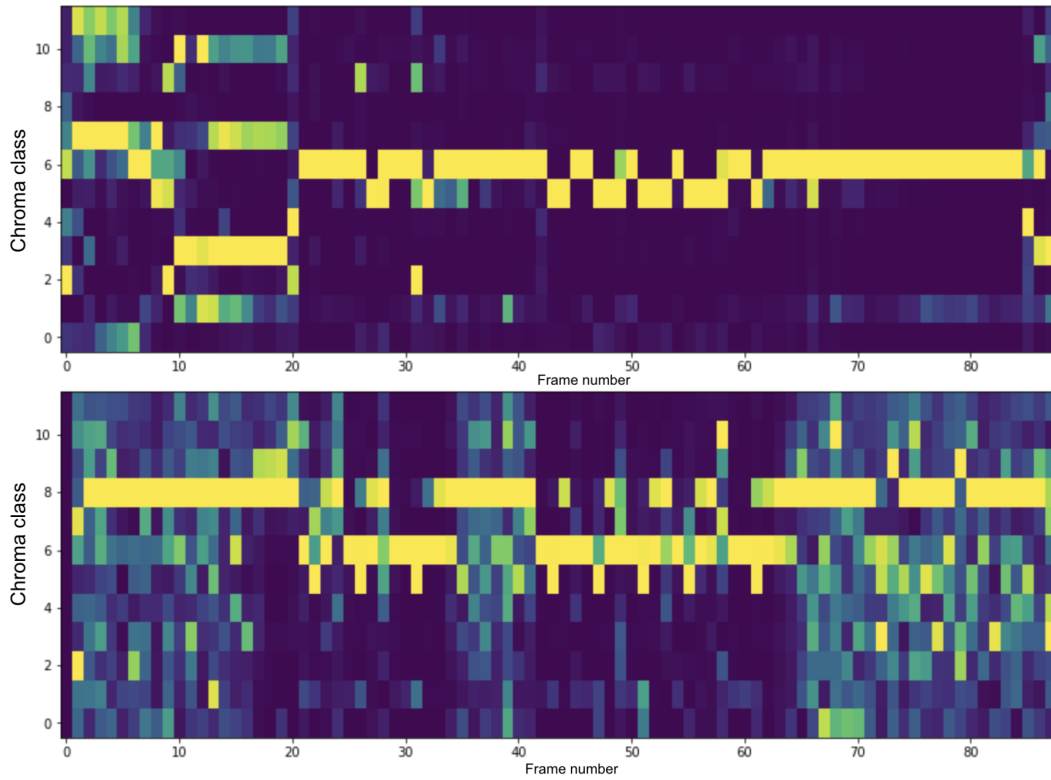
### 4.6.3 Performance Sub-System

The Performance sub-system determines the ways in which short compositions are generated on the fly using the Behavior controls and the *Sprout ideas*.

A selection of *Sprout ideas* are chosen from the vocabulary of your Organism (total collection of available *Sprout ideas*). This selection is made based on the Shy to Excited mode and the 'interesting score' or weight of each *Sprout idea*. In complete 'Shy' mode, the *Sprout ideas* with the highest interesting score are chosen along with phrases of silence. These chosen *Sprout ideas* are then arranged in a sequence of increasing / decreasing or randomized harmonic complexity depending on the chosen envelope style.

Harmonic Complexity is chosen as the variance across the twelve pitch classes in a Chromagram or the Harmonic Pitch Class Profile calculated using the Essentia Library [94]. The Harmonic Pitch Class Profile (HPCP) is computed as a 12-dimensional vector which represents the intensities of

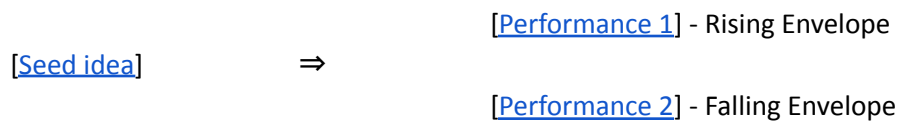
the twelve semitone pitch classes (corresponding to the notes from A to G#) across the spectrum.



**Figure 4.15:** Chromagrams for two different *Sprout* ideas (S0 - top and S1 - bottom) corresponding to the same *Seed* idea. S1 has greater harmonic complexity than S0.

Finally the chosen sequence of *Sprout ideas* is stitched together with a one second fade in and fade out at the beginning and end for each sample, resulting in a short 30-40 second generative composition. These compositions keep changing as the Behavior controls are altered and also as the Vocabulary of Sprouts keeps changing over time with Proliferation.

Here is an example of two distinct performances generated from the same user-submitted *Seed* idea by selecting different envelope styles for each:



This system has been illustrated further here:

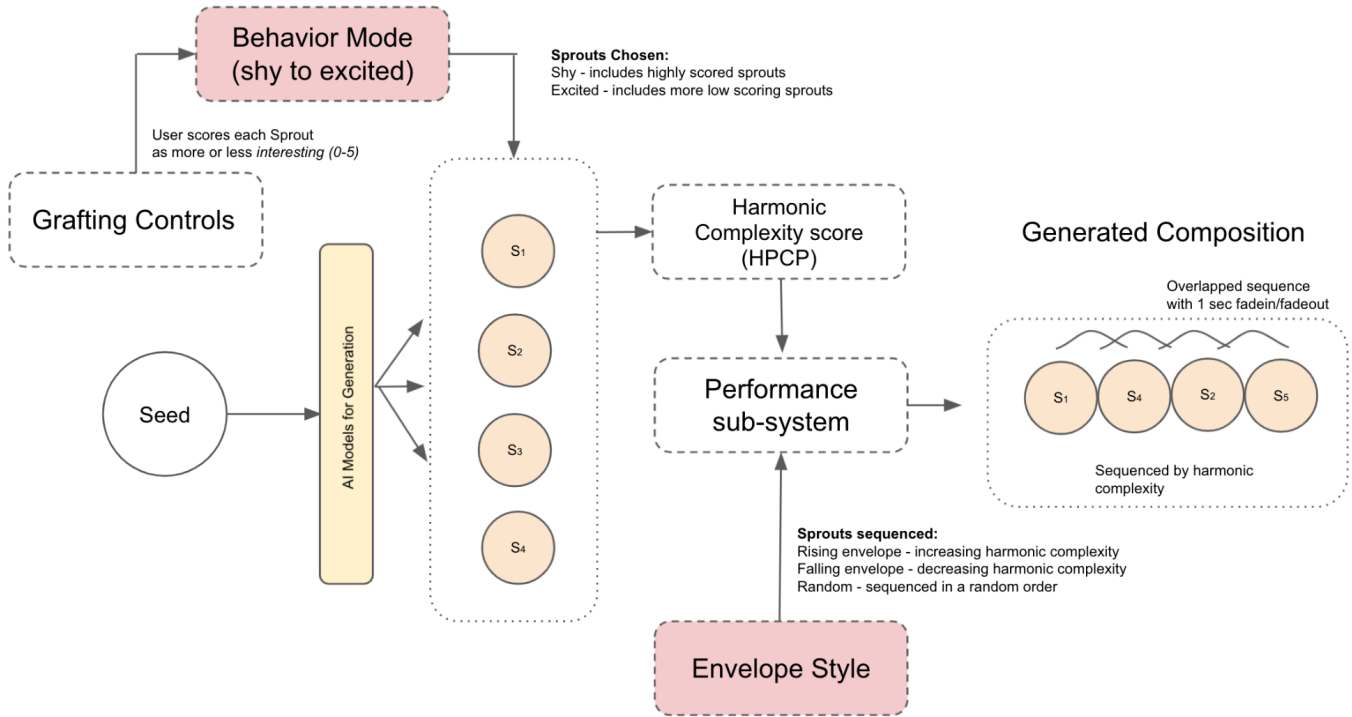


Figure 4.16: System diagram of performance sub-system

## 4.7 Proliferation of your Organism

Once the Organism has been created, shaped and stored in the database. It proliferates overtime by iteratively generating more *Sprout ideas*. These newly generated *Sprout ideas* are ranked according to their similarity to the set of previous *Sprout ideas* with the highest “Interesting score”.

We wish to capture the preferences marked by the user while shaping their Organism, to inform the newly generated *Sprout ideas*. *Sprouts* that are least similar to the old *Sprouts* are deleted automatically. In this way, we create a local knowledge representation of the aesthetic preferences of the user. Similarity is measured by using acoustic features of the audio, described in the section below. Since “interesting score” is a very subjective decision, this is not an exact means of capturing a user’s preferences. But it still proves to be useful to act as a suggestion for new “interesting scores” that can be overridden by the user’s explicit choices as they use the *Grafting* controls on the newly generated *Sprout ideas*.

### 4.7.1 Iterating using Audio Similarity

### Audio Feature Extraction sub-system:

A collection of audio features describing frequency domain properties, loudness properties, rhythmic properties etc. are extracted from each *Sprout idea* using the Essentia Library [94]. This high-dimensional feature space is then used to calculate similarity between two *Sprouts* using a simple euclidean L2 norm.

Here is a list of the acoustic features used to create the high dimensional feature space.

Pitch (YIN)	YIN24 algorithm, with bounds at 200Hz and 1200Hz (and the estimation's confidence).
Pitch (Melodia)	MELODIA74 algorithm, with bounds at 200Hz and 1200Hz (and the estimation's confidence). This is computed as an alternative to YIN that focuses on predominant melody extraction
MFCC	13 coefficients. Essentia's default parameters correspond to the popular MFCC-FB4031 implementation.
Harmonic Pitch Class Profile (or Chroma)	The spectral intensities of each of the 12 pitch classes in twelve-tone equal temperament.
Spectral Centroid	Centroid of the magnitude spectrum, scaled to SampleRate/2
Pitch Saliency	Feature indicating "pitchness" of the signal (from 200Hz to 1200Hz), based on autocorrelation function
Spectral Flux	Euclidean distance between consecutive frames of magnitude spectrum.
Spectral Rolloff	85th percentile frequency of the energy in the magnitude spectrum (i.e. above which 15 percent of the spectrum's energy is)
Spectral Entropy	Shannon entropy of the magnitude spectrum (lower entropy indicates more "peakiness" in the distribution).
Spectral Spread, Skewness, Kurtosis	These measures characterize the shape of the spectral energy distribution, all computed with Essentia's DistributionShape algorithm.
Spectral Contrast	Measure of contrast in each of six sub-bands, by characterizing the distribution shape in each one.
Inharmonicity	The divergence of the peaks in the spectrum from the nearest integer multiples of the estimated fundamental frequency

## 4.8 Key Contributions

- **A brainstorming tool for composers to interact with and query a wide range of AI models with just one's voice.**

We present a brainstorming tool, accessible to different kinds of musicians regardless of musical culture or technical and programming background. A collection of different AI models irrespective of data representation (symbolic or waveform) or computational speed are made interactive in an egalitarian framework that can be queried using one's voice. This enables a composer to generate a collection of sonic material using their voice.

- **A personalized and iteratively proliferating composition object or "Organism".**

Our system focuses on the unique vocalized Seeds and choices made by the user in shaping their Organism's vocabulary and behavior to generate short compositions using the generated material. This provides a personalized composition object that produces contextuality for the AI generated sonic material, leaving room for re-interpretation and provoking serendipity.

# Chapter 5

## Evaluation and Discussion

*“As notions about the nature and function of music become embedded in the structure of software-based musical systems and compositions, interactions with these systems tend to reveal characteristics of the community of thought and culture that produced them.”*

- George Lewis (2000)

*Too Many Notes: Computers Complexity and Culture in Voyager*

This thesis has presented the design and development of *Living, Singing AI* in the form of an accessible, interactive composition tool through a browser application. In this Section we provide a functional qualitative and quantitative analysis of our tool, study the range of generated outputs and describe various usage experiences through the conducted informal peer studies. Our proposed evaluation methods give an insight into the project’s goals and possibility spaces within the context of a selection of application scenarios.

### 5.1 Evaluation Model

Even with the growing research interest in generative systems, the assessment and evaluation of systems that model a space of *creativity*, remain challenging. [95] and [96] describe an evaluation strategy derived from the design ontology of the system. Based on a function - behavior - structure ontology, we evaluate the actual behavior of a system compared to its expected behavior. As the ultimate judge of creative output is the human (composer or listener), subjective evaluations are preferred in this class of generative modeling systems [97].

In the Image generation research community, pattern recognition models to assess and score a generated sample like the *inception score* [98], have been successfully utilised for objective measures of human-like classification abilities [99]. However, the assumed correlation to human judgement still needs further scientific examination, it provides a strong foundation that has been widely adapted. As a contrast, generative music systems offer a much harder problem. The sequential, highly structural, abstract nature of meaning and emotional intent in music [100] make a standardised semantic

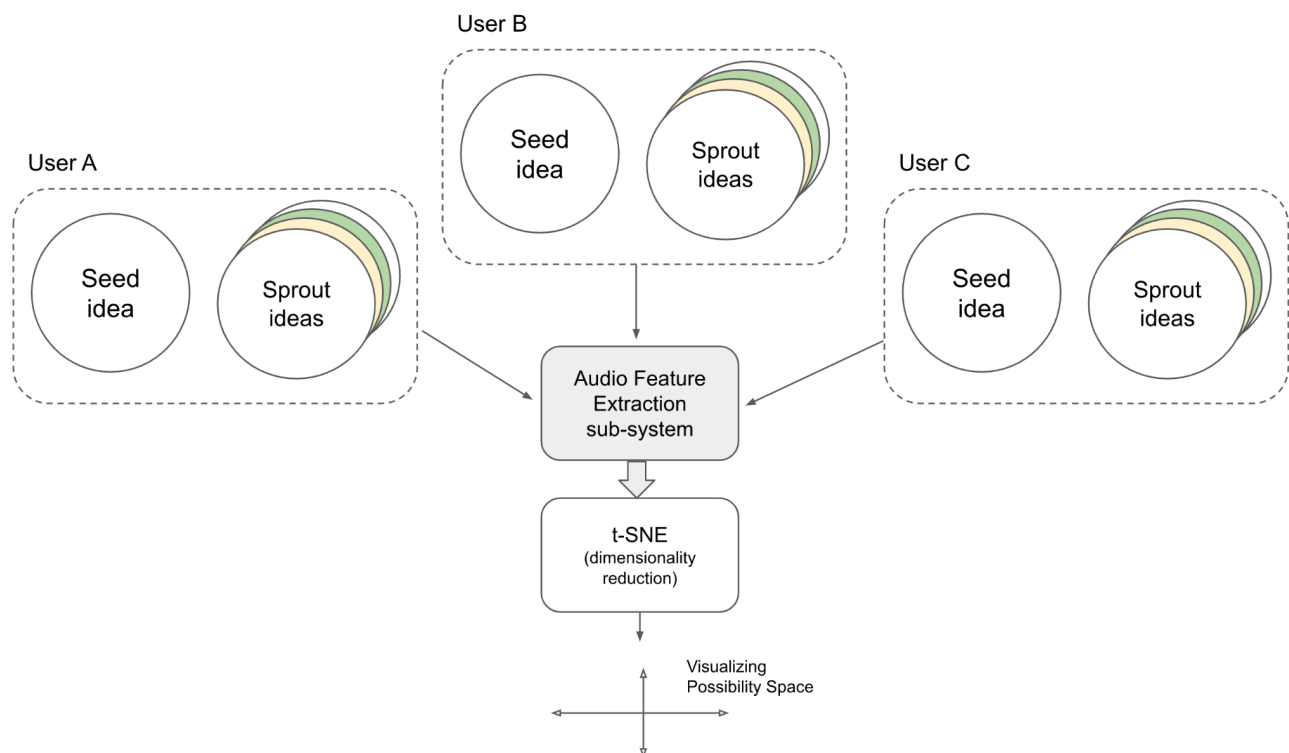
description of music extremely hard. [101], [102] demonstrate why music assessment has not been successfully automated by computational models until now.

Our system's design ontology focussed on the control and range of outputs and its relations to the composer's inputs and expectations provides the necessary constraint for a qualitative and a quantitative functional evaluation.

### 5.1.1 Quantitative

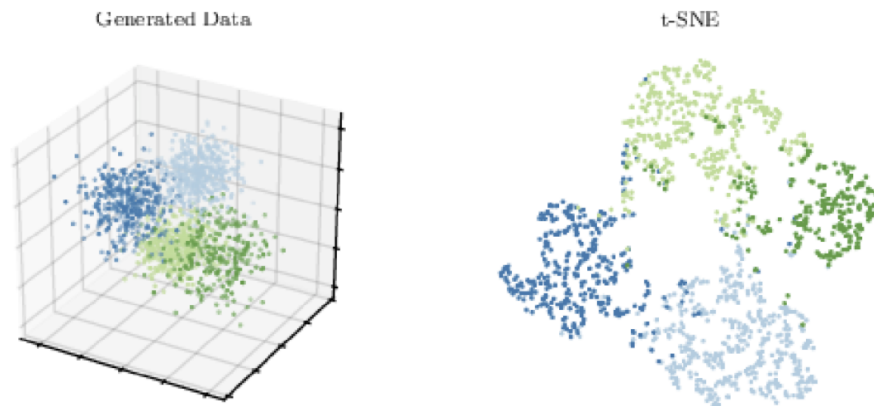
We built a quantitative pipeline for visualizing the *Seed* and *Sprout ideas* generated by each unique user in relation to each other. The field of Music Information Retrieval (MIR) relies on a similar digital signal processing approach to extract information from audio content described as *audio features*. The Essentia library [94] provides a tool for extracting a collection of audio features both time based and frequency based. This set of audio features have been extensively used as a means of an audio fingerprint to visualize and compare audio content similarities in literature from EchoNest [103], AcousticBrainz [65] etc.

Here is a schematic of our Quantitative analysis pipeline, wherein Seed and Sprout ideas from each individual User are passed onto the Audio Feature extraction sub-system. (Described in **Section 4.7.1**), followed by a t-SNE dimensionality reduction step.



**Figure 5.1:** Quantitative Evaluation schematic

t-SNE - visualization [104] is a commonly used statistical method that constructs a probability distribution over pairs of high-dimensional data (in our case the feature sets from two segments of audio), and defines a similar probability distribution in a lower dimensional space by minimizing the KL-divergence between the two distributions.



**Figure 5.2:** Visual representation of dimension reduction through t-SNE

With this pipeline we can visualize the high dimensional representations of our audio examples in a two dimensional semantic space where sampled points that are closer are similar and sampled points that are distanced are dissimilar.

### 5.1.2 Qualitative

We also ran informal peer workshops allowing users to generate *Sprouts* iteratively with our system and reflecting on the range of outputs produced, expectedness and unexpectedness of the results and report some of the key insights discovered below. We demonstrate the ability of our system to generate a rich and diverse set of outputs that are strongly connected to the individual's control inputs.

We collected 'vocalized queries' from 8 musician and 2 non-musician peers to generate and Graft a set of *Sprouts* from each collected Seed, in an informal sound collection study. We use examples from these experiments to functionally analyze our system in the proposed quantitative and qualitative framework.

Further long term qualitative studies focussed on a well defined iterative composition task over time with a diverse group of composers will help support the functional goals of our system. We propose Application scenarios for conducting these tests in a goal-oriented creative scaffolding.



## 5.2 Possibility Spaces

### 5.2.1 Sonic Possibility Spaces

Here we describe an example of a sample user generated *Seed* and a collection of its generated *Sprout* ideas from the three generative algorithms (Jukebox, DDSP and Music Transformer) for the reader.

- A.** User 1 - a collection of Sprouts generated in a 1st iteration from the Initial Seed idea of a user

[\[Initial Seed idea\]](#)

[\[Sprout 1\]](#) [\[Sprout 2\]](#) [\[Sprout 3\]](#) [\[Sprout 4\]](#) [\[Sprout 5\]](#) [\[Sprout 6\]](#) [\[Sprout 7\]](#) [\[Sprout 8\]](#) [\[Sprout 9\]](#)

[\[Sprout 10\]](#) [\[Sprout 11\]](#)

[\[Sprout 12\]](#) [\[Sprout 13\]](#) [\[Sprout 14\]](#) [\[Sprout 15\]](#) [\[Sprout 16\]](#) [\[Sprout 17\]](#) [\[Sprout 18\]](#)

- B.** User 2 - Sprouts arranged in the order *Grafted* by the user with their personal aesthetic preferences

[\[Initial Seed Idea\]](#)

[\[Sprout 1\]](#) > [\[Sprout 2\]](#) > [\[Sprout 3\]](#) > [\[Sprout 4\]](#) > [\[Sprout 5\]](#) > [\[Sprout 6\]](#) > [\[Sprout 7\]](#)

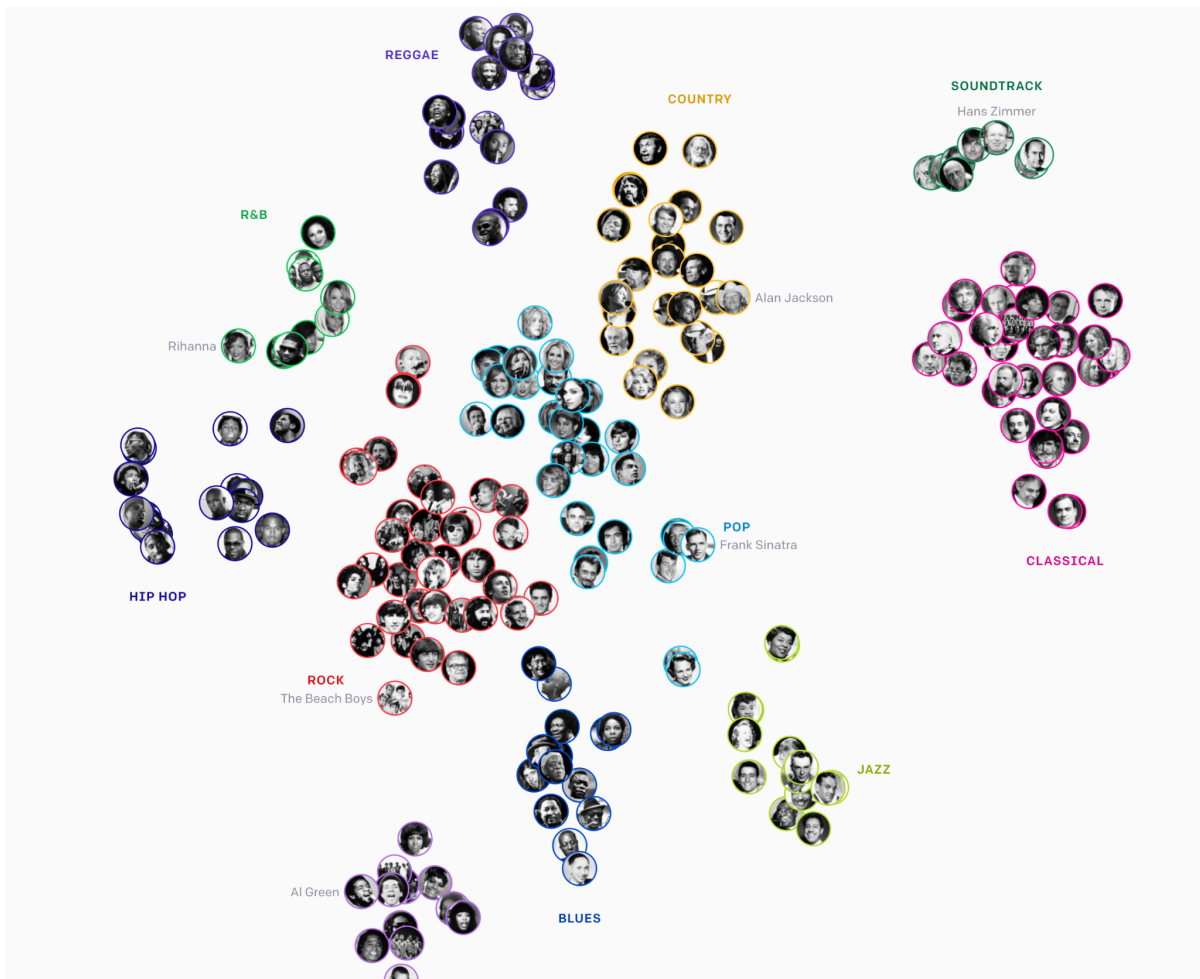
[\[Sprout 8\]](#) > [\[Sprout 9\]](#)

We describe a detailed analysis of the collected Seed ideas and generated Sprout ideas in our Sound Collection Experiment description below. Here we investigate the possibility spaces for each of the three generative models used in our system.

## A. Jukebox

In our quantitative evaluation pipeline, we are able to visualize the diversity of a collection of audio files in a 2-dimensional t-SNE graph. OpenAI provides a t-SNE map of the VQ-VAE codebook vectors learnt from each artist in their training set to demonstrate that similar genres are clustered together.

While this is not directly relevant to our context of generating completely novel sonic material from user defined vocalized queries, it demonstrates the use of t-SNE to quantitatively measure the range of possibilities.

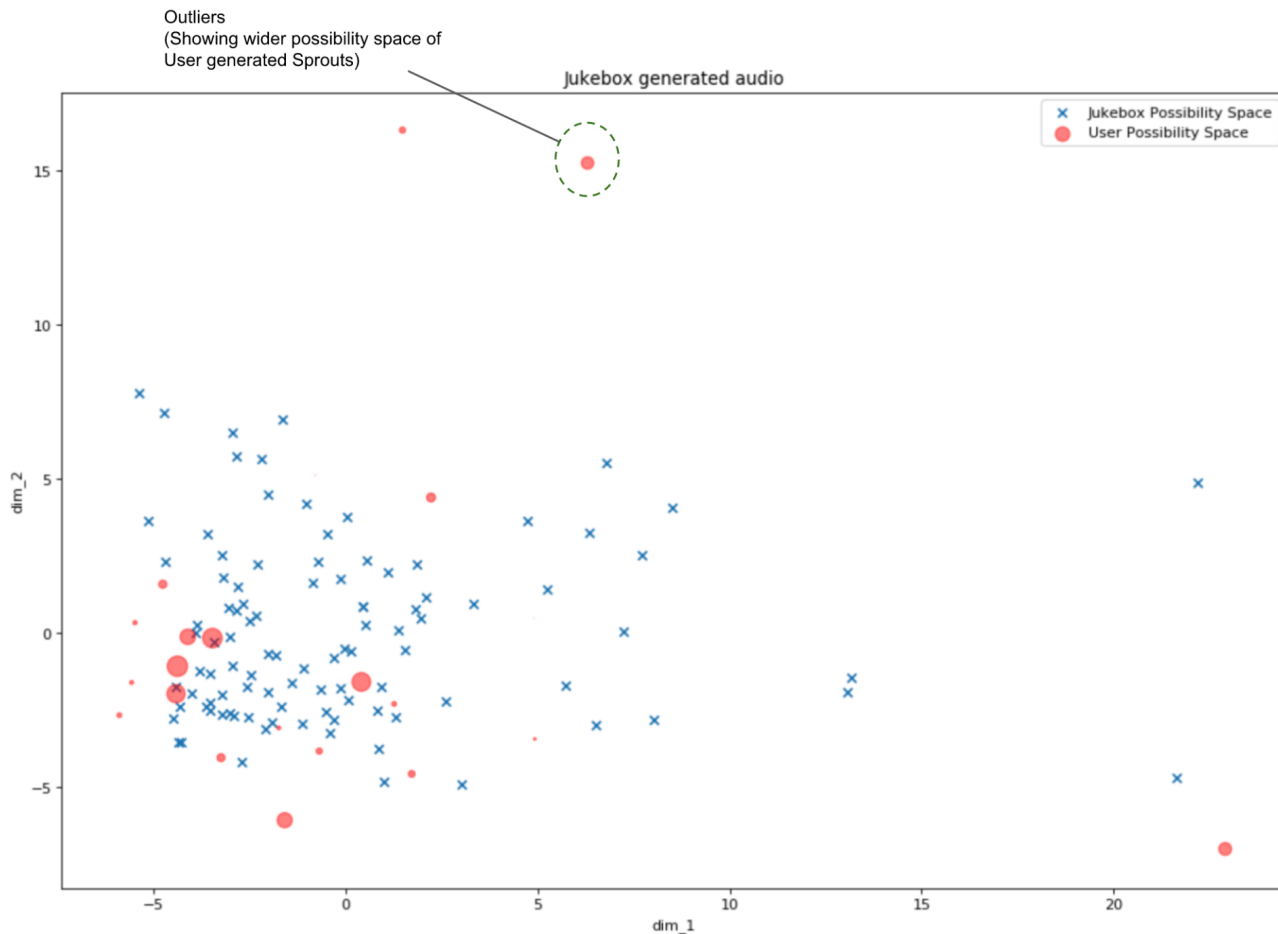


**Figure 5.3:** t-SNE visualization of Jukebox samples from the training set across genres [84]

In order to perform a similar t-SNE visualization on generated audio content from the Jukebox algorithm we scrape the Jukebox soundcloud [105] (only available dataset of official Jukebox examples) to obtain

102 audio continuation samples generated from samples of songs from the training set for example - (Celine Dion, 2Pac, Ella Fitzgerald etc.)

We further collect all the *Sprouts* generated from our user collected vocalized *Seed ideas* and extract their audio feature sets to map to the same t-SNE probability distribution and obtain the range of outputs produced by our system. This is represented in **Figure 5.4**.



**Figure 5.4:** t-SNE visualizations of the Jukebox possibility space.

(Blue) - indicates the audio continuations scraped from official Jukebox examples and  
(Red) - indicates the generated *Sprouts* from user submitted *Seeds* using Jukebox. The size of the red *Sprouts* indicate the interesting scores assigned to the respective *Sprout*.

We find (Figure 5.4) that our *Sprout* Possibility Space, denoted by the red spheres, is as diverse in the t-SNE representation, as the music-primed examples generated by Jukebox, denoted in blue crosses. This indicates that using voice as an input query for priming does not limit the possibility space of outputs that can be generated.

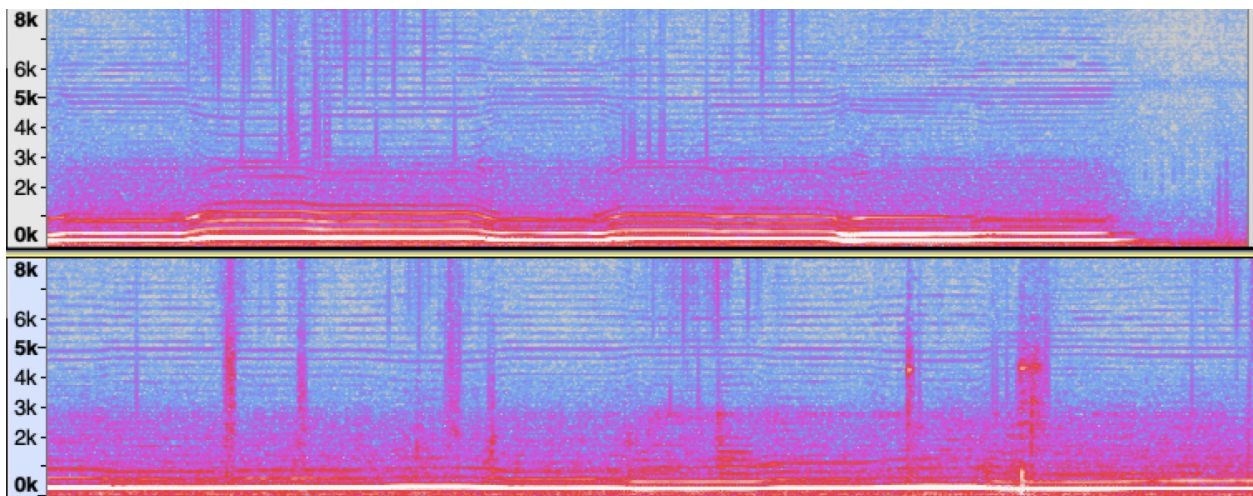
In fact, we observe that some of the *Sprouts* generated by our system (red outliers circled - at the top and also to the right extremes of Figure 5.4) demonstrate outputs that are uniquely novel when compared to the musical continuations (blue) suggesting an even larger diversity of outputs.

Here are examples demonstrating the range of possibilities from the Jukebox system through generated *Sprouts* from user submitted *Seeds*.

- A. User submitted *Seed* containing a vocalized singing example:

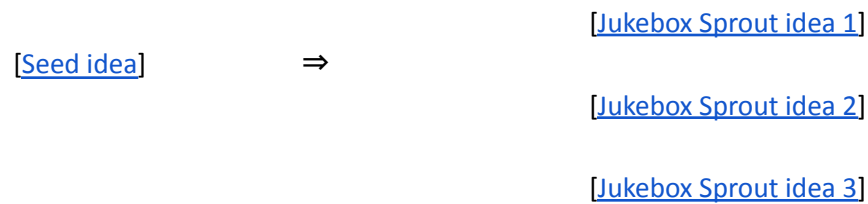


These generated *Sprouts* demonstrate a close relation to the timbres present in the original *Seed*. Both the timbre of the unique singing voice and the lower quality of the microphone at the time of recording are reflected in the generated outputs. In addition to that they demonstrate uniquely new melodic variations realized in the same singing voice.



**Figure 5.5:** Spectrogram representations of (top) Original Seed idea and (bottom) Jukebox generated Sprout idea. Shows shared timbral characteristics including broad spectrum noise under 3kHz. (Audio examples above)

- B. User submitted *Seed* containing a low singing voice that glissandos through an octave and vowel shapes.



In this example we see a really diverse range of *Sprouts* generated.

*Sprout idea 1* maintains a similar timbre and voice quality as the original *Seed* while following a completely novel melodic contour.

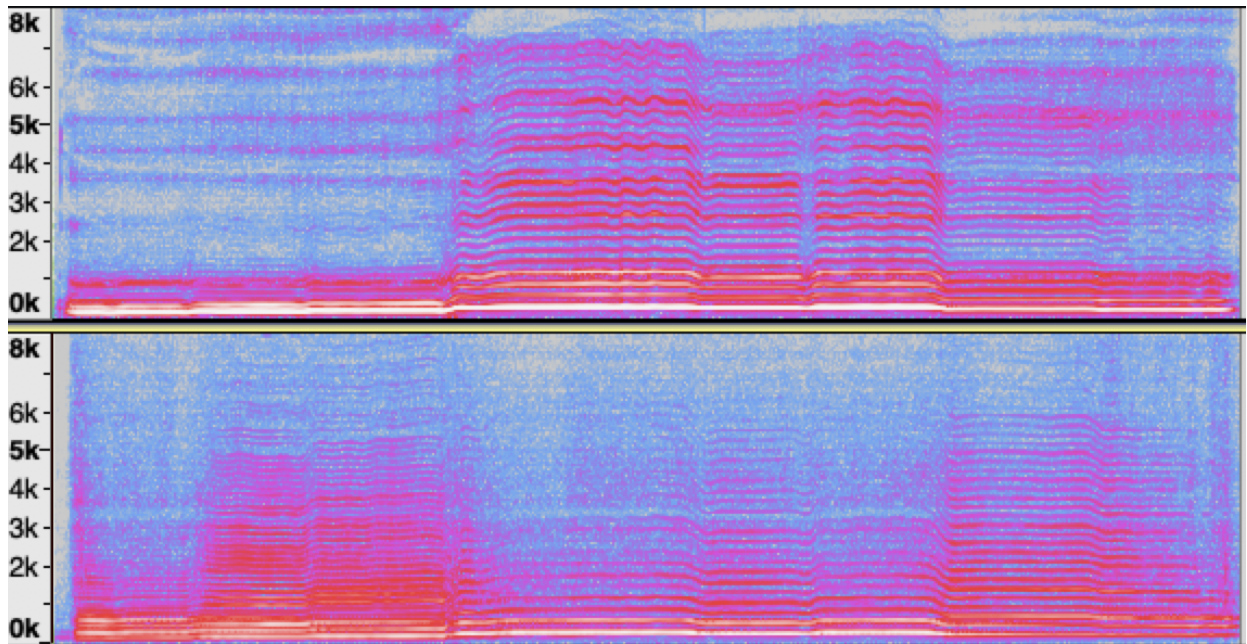
*Sprout idea 2* on the other hand has musically the same key and singing quality but includes a guitar and hi-hat sound mixed together in an almost convincing live performance rendition. This sound is also completely novel and generated at sample level. The voiced lyrics are non-syllables that cycle through different vowel shapes.

*Sprout idea 3* begins with a similar voice quality but quickly spirals out melodically and timbrally, ending with a crescendo of voices and crowd noise.

While the *Sprouts* are diverse they have an internal consistency without any recognizable breaks or auditory glitche effects. All the generated *Sprouts* are also related to the original *Seed* idea in a quality of coarse long range melodic shapes or fine short range timbre details shown in the different examples above.

## B. DDSP

The pre-trained DDSP models that have been made openly available by Google Magenta include the autoencoders trained on individual musical instruments namely - Violin, Trumpet, Tenor Saxophone and Flute.



**Figure 5.6:** Spectrogram representations of (top) Original singing Seed and (bottom) audio generated from DDSP (violin) model as Sprout

While the timbre possibility space is limited to the individual instrument autoencoder models, the pitch contours are strongly conditioned on the pitch contours in the original vocalized query. In our experiments we apply the DDSP instrument filters to some of the Jukebox and Music Transformer generated *Sprouts* to iteratively generate new *Sprouts*. This enables a possibility space that is just as wide as the Jukebox possibility space for each of the instrument autoencoder models available.

Here are some of the wide range of results obtained:

[\[Original Seed idea\]](#) ⇒ [\[DDSP Sprout idea\]](#)

[\[Jukebox Sprout idea\]](#) ⇒ [\[deeper DDSP Sprout 1\]](#)

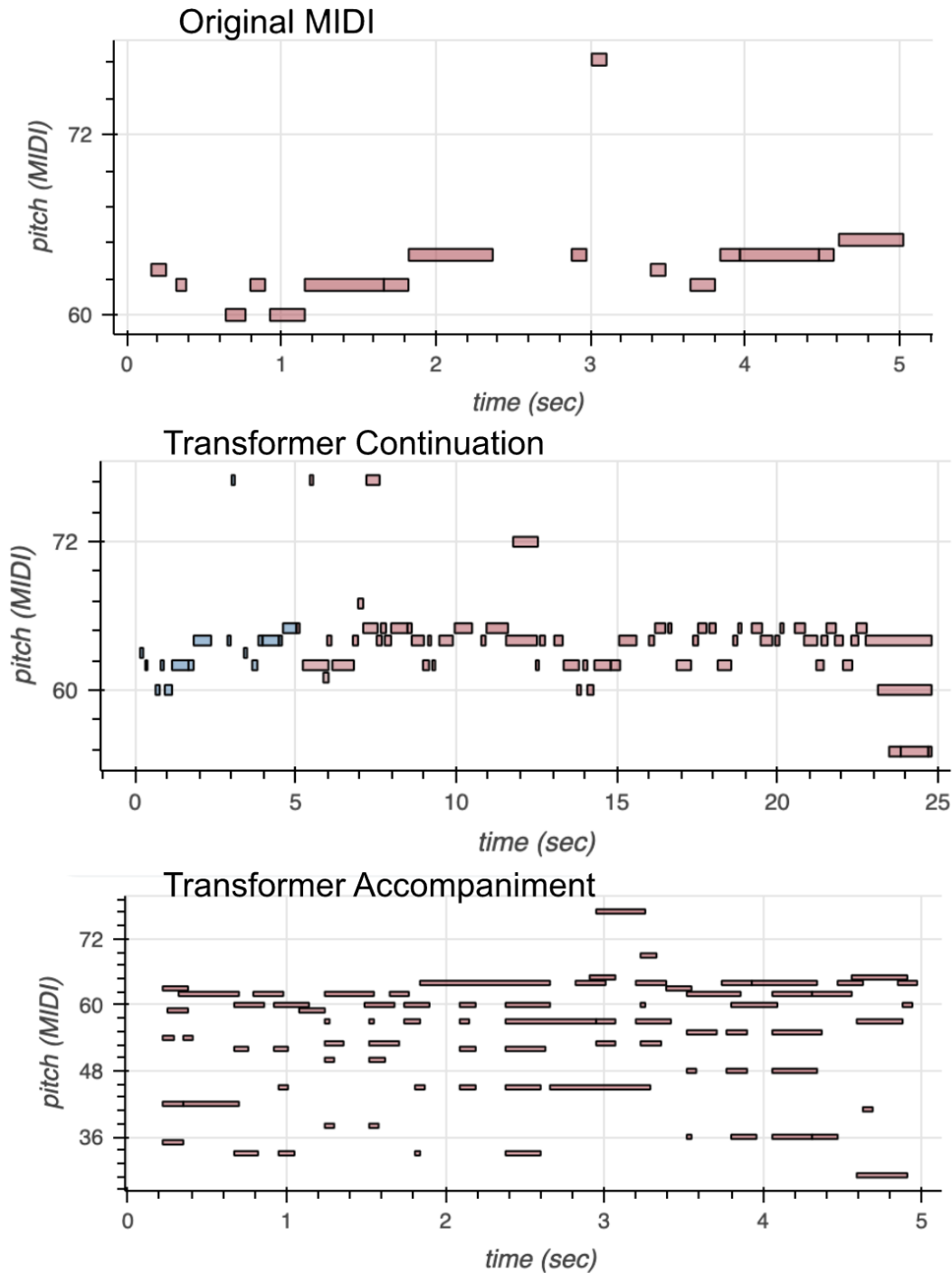
[\[deeper DDSP Sprout 2\]](#)

[\[deeper DDSP Sprout 3\]](#)

## C. Music Transformer

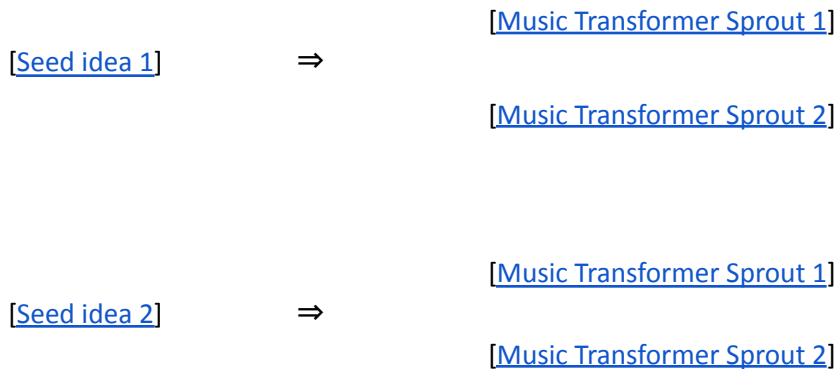
The Music Transformer model acts in the MIDI representation of music and is specifically trained for modeling long term dependencies in piano performances. This does not translate directly to our usage of the tool to generate continuations and accompaniment for vocalized queries. Regardless we explore this language model to create *Sprouts* from the MIDI representations of our widely diverse range of *Seeds*.

Vocal Seeds transcribed as piano MIDI are often chopped into smaller note events, jump between octaves and are forcefully quantized to the 12 notes in an octave. This contrived MIDI leads to non-traditional Music Transformer continuations and accompaniments.



**Figure 5.7:** Music Transformer generated Sprouts in their MIDI representations, (top) - original Seed, (middle and bottom) - generated Music Transformer Sprouts

Here are some Music Transformer generated *Sprouts* from user submitted *Seed ideas*.



## 5.2.2 AI Software Possibility Spaces

Because of the different downstream application goals of individual generative AI software, it is not surprising that a lot of heterogeneity exists in the way the models are formulated, trained and the user interfaces that enable interactions. No standardized methodology exists for evaluating such computational creativity systems.

Exploration refers to the discovery of new resources, knowledge and opportunities and is associated with learning through experimentation. Exploitation on the other hand, refers to the refinement of existing resources and knowledge through explicit control [106]. The degrees of freedom offered by the input method of a given software is directly related to the amount of control it offers to the human user. This capacity of the input to a system can be classified as the dimension of exploitation offered. On the other hand, the range of possible outputs from a generative system, in our case - musical outputs, determines its exploration capacity.

Here we make a functional comparison [95] with the other popularly available AI Music Softwares by identifying the types of input control available and their respective range of output possibilities. It is important to identify that the range of outputs possible are sometimes limited when the software is designed for a very specific application or context - for example video game music or meditative music.

This allows us to qualitatively map the AI Software Space in the chosen exploration and exploitation dimensions. (A complete list of all available AI Software (as on Sept 2021), commercially or privately announced, active or inactive as an exhaustive list in Appendix B.)

### A. Limited Exploration and Limited Exploitation:



AIVA, Jukedeck and Amper:

Jukedeck is one of the oldest commercially available generative music making AI Software tools ideated at the Le Web Conference Paris in 2014. AIVA was founded in Luxembourg in 2016 and Amper Music founded in 2014. While Amper Music has since shifted from supporting individual music makers to launching enterprise products, Jukedeck and AIVA continue to target a broad group of individual composers.

All three of them offer a pre-selection of key, tempo, style and duration of requested music generation. About 10 different styles are offered from rock, pop, electronic, cinematic, modern folk etc. Amper music was specifically designed for adding music to existing videos so it offers a timeline view for creating multiple musical segments. AIVA has the ability to view the generated music in a piano roll form. The instrumentation offered on all three of these systems is a 4-5 track MIDI that can be exported. This very limited means of pre-selecting basic musical structure parameters means the more detailed musical information like melody, rhythm and structure are completely out of control. Also the limited style and instrumentations produce similar sounding outputs on repeated generation.

We categorize these systems as having limited exploration and exploitation possibilities in the class of generative AI music systems.

## **B. Limited Exploration but More Exploitation:**

Humtap, Magenta Studio Plugins:

Humtap was founded in 2013 and many variants like HumOn and HumBeatz have since been available as mobile applications. This system allowed a user to hum a melody and tap a rhythm that was transcribed into symbolic representations. Then the music was generated within a fixed 4-5 track style. This gives greater control over the melodic structure of the music generated but the limitation of style allows only 4-5 timbres. It also does not allow the produced music to be exported as individual MIDI tracks for changing the instrumentation.

Magenta released four of their symbolic recurrent neural network models as Ableton plugins. This was one of the most popular tools amongst the participants of the AI Song Contest (2021). It enables existing MIDI tracks to be imported into the system and then allows for automated drum MIDI to be generated, or creates interpolated new MIDI tracks. This offers both exploration and exploitation strategies. But since all the Magenta plugins are limited to symbolic MIDI domains, they can only be used to generate music symbolically. This does not give access to exploration of novel musical ideas in the waveform domain.

The Magenta plugins however, are a good model for future AI Music tools to be packaged and deployed within existing workflows of musicians.

### C. Slow Exploitation and Directed Exploration: (Our System)

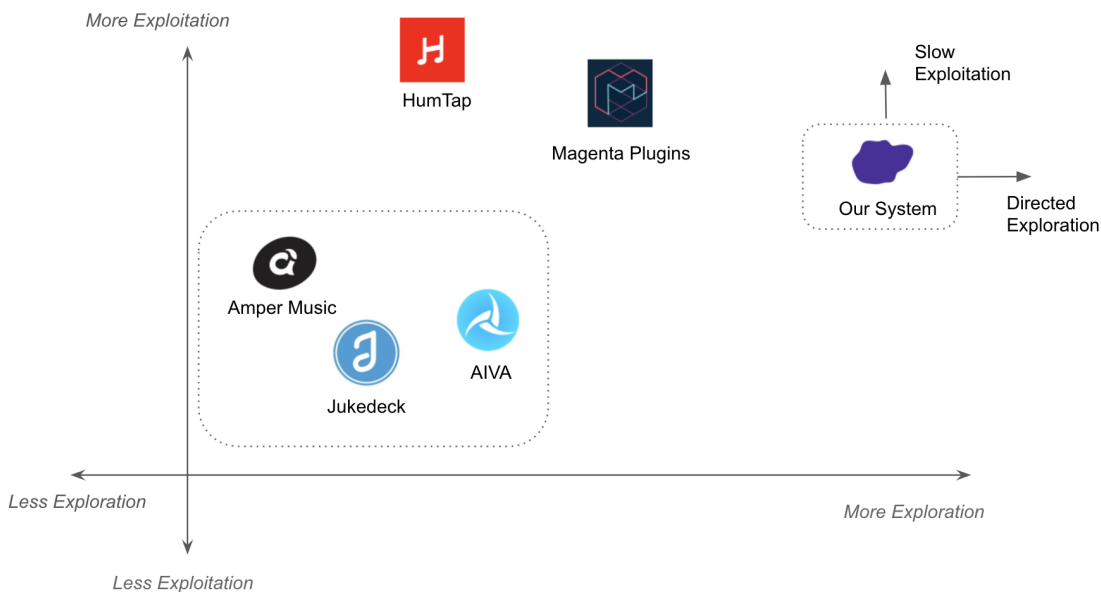
Our system is not intended to produce complete pieces of music like the other softwares enlisted here, with the exception of the Magenta plugins, which also generate short segments of musical phrases.

#### *Slow Exploitation:*

Our system allows the musician to use their voice as an input. The vocalized query is used in its waveform representation and the extracted symbolic MIDI representations to generate novel sonic material. Further exploitation is offered through the *Grafting* and *Behavior* controls. The AI music outputs are iteratively generated and ranked according to the user's aesthetic preferences. This allows for a slow form of exploitation of the mode. The DDSP Sprouts offer complete melodic control to generate new timbres while preserving the pitch information.

#### *Directed Exploration:*

Using voice as a Seed over multiple generative AI models, produces a diverse range of outputs as has been qualitatively described in the previous section. This allows a directed form of exploration through our brainstorming tool. The Jukebox Sprouts and Music Transformer Sprouts both offer great exploration capacity in the waveform and symbolic representation space.



**Figure 5.8:** Visual representation of the AI Software possibility space on an Exploration - Exploitation axes

## 5.3 Sound Collection Experiments

Initial experiments for the *Living, Singing AI* system were done using multiple sound collection stages to explore the range of input interactions and output possibilities of our system.

### Chennai Samples:

In Jan 2020, our team collected sounds in Chennai, India which ranged from children singing, people talking on the streets, Carnatic music instrumental, Carnatic music vocals etc. These pre-recorded sounds were used as a test bed for an early evaluation of our system's generative space.

### Informal Collection from peers:

A set of 'vocalized queries' were collected each, from 8 musicians and 2 non-musician peers which consisted of - (a) spoken utterances, (b) sung melodic phrases, (c) vocalized rhythmic phrases (d) vocal effects and other non-verbal utterances.

## 5.3.1 Musical vs Non-Musical

Here are some examples of Seeds and generated Sprouts from our experiments with musical and non-musical audio queries before collecting vocalized queries from the browser application.

**audio5.1:**

[\[non-music Seed idea\]](#) ⇒ [\[Sprout idea\]](#)

**audio5.2:**

[\[non-music voice Seed idea\]](#) ⇒ [\[Sprout idea 1\]](#)  
[\[Sprout idea 2\]](#)

In audio examples 5.1 and 5.2 we show a field recording of the sounds of vehicles in the city traffic of chennai, and a voice making short bursts of onsets. Both these Seed ideas share the common properties of being without the intention of generating music. In 5.1 our system generates a percussive Sprout with a kind of synthetic drum sound following the rhythmic ideas in the original Seed. In 5.2, *Sprout idea 1* demonstrates a timbrally consistent Sprout following a similar rhythmic characteristic as the voice in the *Seed*. *Sprout idea 2*, on the other hand, takes the same Seed idea and generates a big acoustic percussion sound following the rhythms of the *Seed*.

**audio5.3:**

[\[musical Seed idea\]](#) ⇒ [\[Sprout idea 1\]](#)  
[\[Sprout idea 2\]](#)

**audio5.4:**

[\[singing Seed idea\]](#) ⇒ [\[Sprout idea 1\]](#)

In audio examples 5.3 and 5.4, we show musical *Seeds* and their generated *Sprouts*. In 5.3, we have a solo Cello playing that is continued in *Sprout idea 1* and generates a small segment of a larger orchestration sound in *Sprout idea 2*. In 5.4, a singing voice with a unique timbre is replicated with novel melodic structure.

### 5.3.2 Non-Western Music

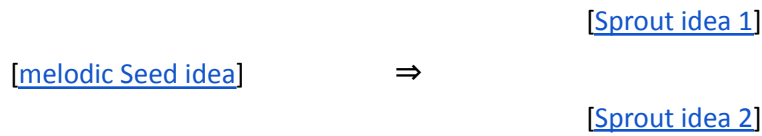
Here we demonstrate the boundaries of our system and the underlying AI models by querying them with non-western musical sources, specifically indian classical music. The training data for our underlying models doesn't include sources of Indian classical music which results in interesting results. We show some of the *Seeds* and *Sprouts* to explain our findings.

In audio example 5.5, the *Seed* contains the rhythmic ideas of a kannokol bol, which is a form of rhythmic solfege. We also hear a drone instrument in the background. In all the generated *Sprouts* we find that the drone is recreated, demonstrating the ability of capturing long term coarse structure. The vocal performance is generated in a unique non-syllabic phrasing that replicates the style of the original *Seed*. In *Sprout idea 3*, we demonstrate the underlying MIDI methods' ability to generate a piano performance unlike human piano playing but capturing the audio content of the *Seed*.

**audio5.5:**

[\[rhythmic Seed idea\]](#) ⇒ [\[Sprout idea 1\]](#)  
[\[Sprout idea 2\]](#)  
[\[Sprout idea 3\]](#)

**audio5.6:**

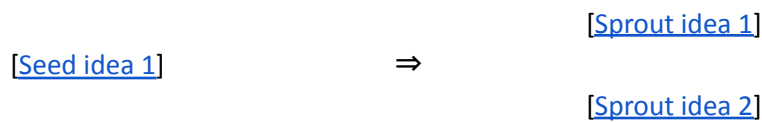


In audio example 5.6, we have three elements, a drone instrument in the background, a percussive accompaniment and the main vocal performance of Indian classical music. Since the underlying models have not been trained on Indian classical music, they capture more fundamental musical properties like rhythm and pitch realized in different instrumentations. These are shown in the respective *Sprout ideas* generated.

### 5.3.3 Non-Verbal Utterances

Here we demonstrate a collection of non-verbal *Seeds*. Audio example 5.7 has a whispering sound that is reflected in its *Sprout idea 1*. While *Sprout idea 2*, transforms the whispering sound into a drum break. Audio example 5.8 has a simple beat-boxing utterance that is transformed into a generated instrumental *Sprout*. Audio example 5.9 has a wah-pedal-like vocal effect of changing vowel shapes that is also replicated in the generated *Sprout* with a unique melodic contour. *Sprout idea 2* demonstrates a change in the timbre maintaining the melodic contours of the original *Seed*.

**audio5.7:**



**audio5.8:**



**audio5.9:**



[\[Seed idea 3\]](#)

⇒

[\[Sprout idea 2\]](#)

### 5.3.4 Seed to Composition

We also present a collection of the Seed and generated Sprouts arranged into the computationally generated compositions. These compositions contain various Sprouts generated from each Seed that have been shaped by the *Grafting* and *Behavior* controls by the unique composer that created the *Seed ideas*. This demonstrates a range of output Sprouts with added contextuality demonstrating our system’s ability to be used as a brainstorming instrument.

**audio5.10:**

[\[Seed idea 1\]](#)

⇒

[\[Composition\]](#)

**audio5.11:**

[\[Seed idea 2\]](#)

⇒

[\[Composition\]](#)

**audio5.12:**

[\[Seed idea 3\]](#)

⇒

[\[Composition\]](#)

**Audio5.13:**

[\[Seed idea 4\]](#)

⇒

[\[Composition\]](#)

### 5.4.5 Summary

In this section, we presented an evaluation model for quantitatively measuring the possibility space of AI generated Sprouts from our system and qualitatively analyzing the functional goals of our system and the underlying generative AI models. We also presented a discussion on commercially available AI music software systems and situated our proposed system on the axes of exploration (range of generated outputs) and exploitation (flexibility of inputs).

Our sound collection study demonstrated that vocalized queries can be used as a flexible form of input that allows a range of musical ideas - sung melodies, rhythmic phrases, non-verbal utterances,

beat-boxing, breathy stochastic noise and miscellaneous vocal effects. The generated Sprouts had the ability to create timbrally consistent material with novel melodic and rhythmic content as well as new timbres maintaining the melodic and rhythmic information of the Seed. We demonstrated these characteristics of our system with a collection of user submitted Seed examples and their respective generated Sprouts through our browser application.

Further experiments with a larger diversity of composer generated vocalized queries can help identify the limitations and boundaries of each of the underlying AI model possibility spaces. We presented an audio-content-based t-SNE visualization for this objective. Alternate forms of visualizing the high dimensional latent representations will help us compare the underlying model possibility spaces in more rigorous ways. We also observe that composer focused evaluation and scoring of generated material regardless of AI model, allowed for an egalitarian brainstorming tool. Our study proposes an evaluation framework that can be extended to longer time scales, where participating composers can use the system for specific composition tasks with iterative reflection over the generated material. The ease of running browser-based experiments, observed in our early surveys, also demonstrates the accessible nature of our system towards non-programming composers.

Our discussion, supported with chosen examples from our system, help demonstrate: the range of generated outputs (exploration) and flexibility of inputs (exploitation) - axes of our proposed brainstorming composition system.

# Chapter 6

## Conclusion

*“New means change the methods, new methods change the experience,  
and new experiences change the man.”*

- Karlheinz Stockhausen  
(1977, lecture on electronic music)

This thesis has presented the design and development of an interactive non-programming environment for accessible interaction and personalization of modern AI music generation models with just one’s *voice*. As part of our work, we examined the rich source of algorithmic music composition and new paradigms of AI tools in creativity. We analysed the space of presently available generative music making tools and their limitations in practice motivating the key principles of accessibility and personalization that guided our work. We further qualitatively and quantitatively identified the possibilities and limitations of a functional approach to our proposed system.

### 6.1 Contributions

The formulation of the *Living, Singing AI* system and the accompanying discussion presented in this document offer the following contributions:

- A novel *Voice-based* interaction methodology for the design of flexible and accessible AI Music Softwares.

This form of input medium for interaction enables a flexible stream of querying the system that can be used to exploit the representations learnt and explore its intended and unintended boundaries. Systems intended for assisting a creative goal benefit from allowing their misuse. We also demonstrate that voice as an input, makes computational methods for creativity accessible to a wide range of non-programming musicians.



- An egalitarian framework for multiple generative AI Music models to be evaluated at the level of the composer. This enables the same input queries to be tested across multiple generative models with a focus on the sounds generated rather than the methods that produced them.
- An adaptable generative music instrument scheme that encourages reflection at each step of interaction resulting in a bespoke system. Our system presents an example of the unique ability of AI musical instruments to *learn* from the aesthetic preferences of a unique individual. We propose a system centered around the individual composer that can be intentionally directed.
- A brainstorming tool to provoke and direct serendipity and short compositions that add contextuality for provocation leaving room for re-interpretation.

## 6.2 Future Work

### A. Long-Form Composer Study:

In this work, we have presented initial studies inviting composers to interact and reflect using our system without the specific goals of intentional composing. As an immediate next step we propose a long-form study with a collection of diverse composers that use the system over a period of time to accomplish a predefined musical goal.

We can adapt this system to a city-wide symphony composition like those proposed by Tod Machover [107],[108]. This model would invite music-makers from diverse musical cultures at different technological and musical skill levels to use our system over multiple time scales. This will enable us to test the directed brainstorming goals and the aesthetic limitations of its use over time. Our system is not intended to be a replacement for usual composition and music-making workflows but rather to perform a specific goal of generating personalized sonic material. A long form composer study would validate its use within the existing workflows of musicians.

### B. Community Spaces:

Our system allows for a composer to export their generated sonic material into a local copy of audio and MIDI files that can be used in other audio workstations. An immediate goal would be to extend this export functionality of the system by creation of a share-able community space where composers can export / share / release their *AI music Organisms*. Allowing for remixing and iterating in community spaces would enable a richer interaction with our system and with individual sonic ideas. Allowing

vocalized queries that are currently personal to be shared amongst a community would enable new research directions of data collectives or audio sample collectives.

### **C. Method Deployment:**

We have demonstrated the use of an egalitarian system with three chosen generative AI music methods. A long term goal would be to enable deployment of new generative music models into the existing *Living, Singing AI* system. A community of creators familiar with our system would be able to immediately access newer models. Academics creating new models for music could also test and validate their model with an existing network of creators.

### **D. New Music:**

The early adaptation of AI tools in the visual arts and the rapidly growing ecosystem of creators, novel methods of distribution and consumption are indicators of the exciting future of AI Music. Artistic forms and narratives that accompany the affordances of AI as a creative tool unlock new paradigms of who is a composer and what constitutes a composition.

In this thesis we have shown examples of mutable generative compositions created completely from the chosen generative models starting with a single *Seed* of a personal vocalized query. A long term-goal would be to drive forward the research into new forms of AI music in the following dimensions:

#### *Form Factors:*

Just as the browser based, non-programming interface gave rise to certain directions for the presented accessible system, newer form factors of deploying AI models would explore new aesthetics of control and performance. A hardware implementation with buttons, knobs and patch cables would fit into an extensive ecosystem of hardware and modular synthesizers. Newer tangible and moldable material forms would provoke newer interpretations of querying and interacting with AI models as musical instruments.

#### *Democratized and Decentralized Music:*

AI powered and assisted musical instruments have the potential of defining a new music aesthetic, which needs to be observed as they get assimilated into society. As these tools are deployed, the levels of democratization of music datasets and methods will determine the directions these assimilations might take. New forms of licensing for share-able data collectives and AI generated content will also

determine the state of this new music. Our system proposes an experiment in technologically democratizing access to already publicly available generative methods but further work needs to be done on firmly grounding ownership and credit assignment through standardized methods. A decentralized version of a DAO producing generated music with shared data and music ownership is one such system.

By proposing a new system for deploying, accessing and interacting with AI music methods, this thesis hopes to explore new directions of AI Music that are centered around inspiring and assisting the goals of the individual human composers that use them.

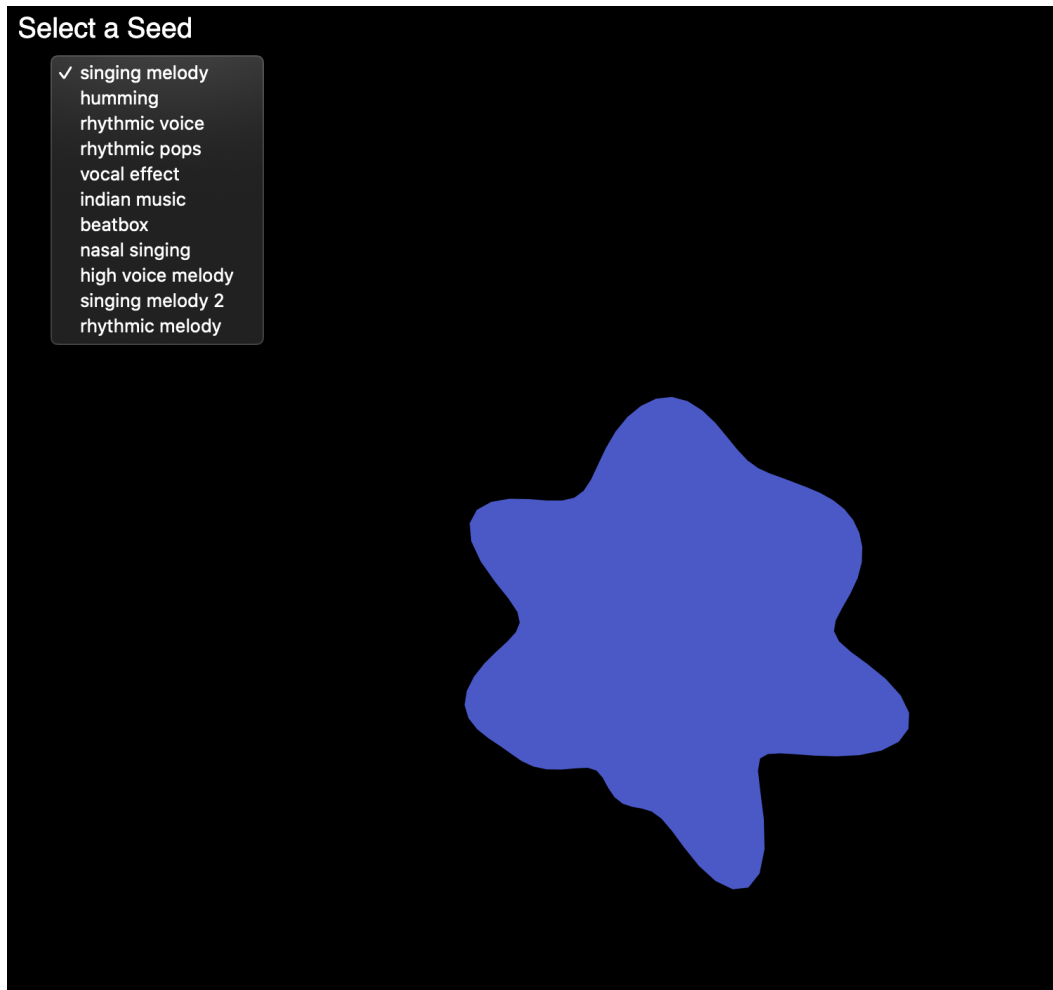
Our work in developing human-centered AI musical tools imagines a future where the mathematical ideas of AI serve to inspire and assist in newer paradigms of the human creative process. Early use of AI for imitating styles and patterns from musical datasets and canons will soon be replaced by the more intimate human functions of music - for communicating, story-telling and reflection. These future AI musical instruments, in the creative hands of a unique individual, could act as portals, liberating sonic worlds across time and cultures, in newer paradigms of authorship.

While AI ideas have the ability to capture a representation of a piece of reality through the datasets that they are trained on, AI tools like the proposed future musical instruments, allow an individual to react and reflect on this reality. To be human is to be *a* human with a unique set of experiences and opinions. We look towards a future where AI musical instruments can be molded and shaped in the hands and ideas of an individual - be it a composer, a folk story-teller or a mother singing a lullaby.

# Appendix A

Sound examples generated from the *Living, Singing AI* system:

**Organisms** (generated by participating users and field Sound Collections):



Here is an Interactive Browser version of this Appendix A, to explore a gallery of Organisms generated from our *Living, Singing AI* system.

[https://web.media.mit.edu/~manaswim/Thesis\\_Media/Demos/organismGallery/](https://web.media.mit.edu/~manaswim/Thesis_Media/Demos/organismGallery/)

1. User 1: ***Singing Melody***

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

[\[Sprout 11\]](#)

2. User 2: ***Humming***

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

[\[Sprout 11\]](#)

3. User 3: ***Rhythmic Voice***

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

[\[Sprout 11\]](#)

[\[Sprout 12\]](#)

4. User 4: ***Rhythmic Pops***

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

[\[Sprout 11\]](#)

5. User 5: **Vocal Effect**

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

6. User 6: **Indian Music**

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

7. User 7: **Beat-box**

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

8. User 8: **Nasal Singing**

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

9. User 9: **High Voice Melody**

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

[\[Sprout 11\]](#)

[\[Sprout 12\]](#)

10. User 10: **Singing Melody 2**

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

[\[Sprout 9\]](#)

[\[Sprout 10\]](#)

[\[Sprout 11\]](#)

11. User 11: **Rhythmic Melody**

[\[Seed idea\]](#)

[\[Sprout 1\]](#)

[\[Sprout 2\]](#)

[\[Sprout 3\]](#)

[\[Sprout 4\]](#)

[\[Sprout 5\]](#)

[\[Sprout 6\]](#)

[\[Sprout 7\]](#)

[\[Sprout 8\]](#)

## Appendix B

List of AI Music Software:

Company / Tool Name	Corporation / Startup / Research	Location	Description
MAIA Suggest	Research startup	Davis, California, U.S.	Drop-down menu for style, tempo. <a href="https://maia-suggest.glitch.me/">https://maia-suggest.glitch.me/</a>
AIVA	startup	Luxembourg	video game, movie and commercial music generation <a href="https://www.aiva.ai/">https://www.aiva.ai/</a>
Magenta Studio	Corporation (Google)	S.F./Bay area, U.S.	Collection of ableton live plugins <a href="https://magenta.tensorflow.org/studio/">https://magenta.tensorflow.org/studio/</a>
Magenta PerformanceRNN	Corporation (Google)	S.F./Bay area, U.S.	Symbolic piano performance modeling - velocity and timing <a href="https://magenta.tensorflow.org/performance-rnn">https://magenta.tensorflow.org/performance-rnn</a>
Magenta MusicVAE	Corporation (Google)	S.F./Bay area, U.S.	Symbolic variational autoencoder (4 tracks) <a href="https://magenta.tensorflow.org/music-vae">https://magenta.tensorflow.org/music-vae</a>
Magenta Onsets and Frames	Corporation (Google)	S.F./Bay area, U.S.	Waveform to MIDI trained on piano performances <a href="https://magenta.tensorflow.org/onsets-frames">https://magenta.tensorflow.org/onsets-frames</a>
Magenta Coconet	Corporation (Google)	S.F./Bay area, U.S.	BACH 4 part harmony generation. <a href="https://github.com/magenta/magenta/tree/main/magenta/models/coconet">https://github.com/magenta/magenta/tree/main/magenta/models/coconet</a>
Magenta Music Transformer	Corporation (Google)	S.F./Bay area, U.S.	Transformer Model for piano. <a href="https://magenta.tensorflow.org/piano-transformer">https://magenta.tensorflow.org/piano-transformer</a>
Magenta DDSP	Corporation (Google)	S.F./Bay area, U.S.	Differentiable Digital signal



			processing - modular autoencoders in a sinusoidal modeling framework. <a href="https://sites.research.google/tonetransfer">https://sites.research.google/tonetransfer</a>
OpenAI MuseNet	Research org	S.F./Bay area, U.S.	Symbolic deep neural network generating 4 min compositions with 10 instruments with style conditioning. <a href="https://openai.com/blog/musenet/">https://openai.com/blog/musenet/</a>
OpenAI Jukebox	Research org	S.F./Bay area, U.S.	Sample based sparse transformer from hierarchical VQ-VAE trained on 1.2million songs (undisclosed dataset scraped from web)
PopGun	Artists	Australia	Pick Genre, mood, duration - trained on collection of electronic samples.
Amper (bought by Shutterstock - 11/11/2020)	startup	New York, U.S.	Cloud based music generation from style, mood and duration. Paid service.
Alysia (WAVEAI)	startup	S.F./Bay area, U.S.	Singing AI mobile application, Karaoke.
AUDOIR SAM	startup	S.F./Bay area, U.S.	SAM - a suite of songwriting tools to write melodies, chords, lyrics. <a href="https://www.audoir.com/how-to-build-a-songwriting-ai">https://www.audoir.com/how-to-build-a-songwriting-ai</a>
LALA.AI	Commercial service		Source Separation/ Extracts vocal and instrumental tracks from audio (browser service)
Humtap	startup	Palo Alto, U.S.	Voice based AI-Music mobile application
LANDR	startup	Montreal, Canada	Cloud based mastering services

Flow Machines	Corporation (SONY)	Paris, France	Markov Chain models for generating melodies. Daddy's Car - in the style of the Beatles.
Melodrive	startup	Bay area, U.S.	Generates video game music in real-time for endless soundtracks.
Jukedeck (bought by TikTok 2018)	Corporation (TikTok)	London, U.K.	Generates symbolic music melody and harmony - from selection of style, mood, duration.
IBM Watson Beat	Corporation (IBM)	New York, U.S.	Cloud based app for symbolic music generation from mood and primed MIDI sequences.
Endel	Startup	Berlin, Germany	Generates personalized soundscapes for mindfulness
Dadabots (SampleRNN)	Artists	U.S. / Berlin	SampleRNN model adapted to death metal, free jazz. <a href="https://github.com/dada-bots">https://github.com/dada-bots</a>
Morpheus	Research group	U.S.	Hybrid Machine Learning models to generate structured music.
Ross Goodwin / YACHT	Artist	U.S.	Lyric generation algorithm <a href="https://rossgoodwin.com/">https://rossgoodwin.com/</a>
Magenta NSynth Super	Corporation (Google)	S.F./Bay area, U.S.	Sample based timbre generation, 16 dimensional latent space. Hardware instrument
Sony CSL	Corporation (SONY)	Paris, France / Japan	DeepBach, RNN and variation networks for style imitation
Amazon AWS Deep Composer	Corporation (Amazon)	U.S.	Hardware implementation synthesizer. Supports training custom GAN models.
Voyager	Artist / Research	U.S.	Improvisation through machine listening for free

			jazz improvisation.
David Cope	Artist / Research	UC Santa Cruz, U.S.	EMI - rule based pattern grammar built for specific composers -
Brian Eno 'Koan' program	Artist / Research	U.K.	Generative music systems from SSEYO as standalone players and browser plugins.

# References

[1] "Al-Khwarizmi biography"

<https://mathshistory.st-andrews.ac.uk/Biographies/Al-Khwarizmi/>

Archived from the original on August 2, 2019

[2] George Brecht's Drip Music (1962) at MoMA

<https://www.moma.org/collection/works/127311>

[3] Maconie, Robin. "Stockhausen's 'Setz die Segel zur Sonne'." Tempo 92 (1970):

30-32. <https://www.jstor.org/stable/943182>

[4] Xenakis, Iannis. Formalized music: thought and mathematics in composition. No. 6. Pendragon Press, 1992.

[5] Perkis, Tim, et al. The League of Automatic Music Composers, 1978-1983. New World Records, 2007.

[6] Lewis, George E. "Too many notes: Computers, complexity and culture in voyager." Leonardo Music Journal (2000): 33-39.

[7] Legg, Shane, and Marcus Hutter. "A collection of definitions of intelligence." Frontiers in Artificial Intelligence and applications 157 (2007): 17.

[8] <http://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/>

(accessed on Aug 13th, 2021)

[9] Ross Goodwin's Spotlight

<https://blog.library.tc.columbia.edu/b/18716-Learning-Theater-Spotlight-Expanding-the-Potential-of-Written-Language-with-AI-Bots>

(accessed on Aug 13th, 2021)

[10] AI art 'Portrait of Edmond Belamy' sold:

<https://www.nytimes.com/2018/10/25/arts/design/ai-art-sold-christies.html>

(accessed on Aug 13th, 2021)

[11] Generative Adversarial Network art

<https://towardsdatascience.com/gangogh-creating-art-with-gans-8d087d8f74a1>

(accessed on Aug 13th, 2021)

[12] Hadjeres, Gaëtan, François Pachet, and Frank Nielsen. "Deepbach: a steerable model for Bach chorales generation." International Conference on Machine Learning. PMLR, 2017.

[13] Sturm, Bob, Joao Felipe Santos, and Iryna Korshunova. "Folk music style modelling by recurrent neural networks with long short term memory units." 16th International Society for Music Information Retrieval Conference. 2015.

[14] Krumhansl, Carol L., and Lola L. Cuddy. "A theory of tonal hierarchies in music." Music perception. Springer, New York, NY, 2010. 51-87.

[15] <https://neurips2020creativity.github.io/>

(accessed on 10th Aug, 2021)

[16] <https://boblsturm.github.io/aimusic2020/>

(accessed on 10th Aug, 2021)

[17] <https://www.aisongcontest.com/>

(accessed on 10th Aug, 2021)

[18] AI Song Contest, Netherlands (VPRO) - Report

<https://magenta.tensorflow.org/aisongcontest>

(accessed on 10th Aug, 2021)

[19] Tsay, Jason, et al. "Runway: machine learning model experiment management tool." Conference on Systems and Machine Learning (SysML). 2018.

[20] Coggan, Melanie. "Exploration and exploitation in reinforcement learning." Research supervised by Prof. Doina Precup, CRA-W DMP Project at McGill University (2004).

[21] Landy, Leigh. "Sound-based music 4 all." The Oxford handbook of computer music. 2009.

[22] [https://johncage.org/pp/John-Cage-Work-Detail.cfm?work\\_ID=30](https://johncage.org/pp/John-Cage-Work-Detail.cfm?work_ID=30)

(John Cage's ASLSP, accessed on Aug 15th, 2021)

[23] <http://news.bbc.co.uk/2/hi/europe/7490776.stm> - Article on longest piece of music

(accessed on Aug 5th, 2021)

- [24] <https://www.daphneoram.org/oramicsmachine/>  
(Daphne Oram's 'The Oramics Machine' - accessed on Aug 15th)
- [25] Agres, Kat, Jamie Forth, and Geraint A. Wiggins. "Evaluation of musical creativity and musical metacreation systems." *Computers in Entertainment (CIE)* 14.3 (2016): 1-33.
- [26] Perkis, T., Bischoff, J., Horton, J., Gold, R., DeMarinis, P., & Behrman, D. (2007). *The League of Automatic Music Composers, 1978-1983*. New World Records.
- [27] G. E. Lewis. Too many notes: Computers, complexity and culture in Voyager. *Leonardo Music Journal*, 10:33–39, 2000.
- [28] Kenmochi, Hideki, and Hayato Ohshita. "Vocaloid-commercial singing synthesizer based on sample concatenation." Eighth Annual Conference of the International Speech Communication Association. 2007.
- [29] Hedges, Stephen A. "Dice music in the eighteenth century." *Music & Letters* 59.2 (1978): 180-187.
- [30] Hiller, Lejaren Arthur, and Leonard M. Isaacson. *Experimental Music; Composition with an electronic computer*. Greenwood Publishing Group Inc., 1979.
- [31] <http://artsites.ucsc.edu/faculty/cope/experiments.htm>  
David Cope's comments on EMI (accessed on 15th Aug)
- [32] Cope, David. "Computer modeling of musical intelligence in EMI." *Computer Music Journal* 16.2 (1992): 69-83.
- [33] <https://toplap.org/wiki/ManifestoDraft>  
Manifesto for Algorithmic Music Performance (accessed on 15th Aug)
- [34] Collins, Karen. "An introduction to procedural music in video games." *Contemporary Music Review* 28.1 (2009): 5-15.
- [35] Cragg, Michael. "Björk's Biophilia." *The Guardian* 28 (2011).
- [36] Goodfellow, Ian. "Nips 2016 tutorial: Generative adversarial networks." arXiv preprint arXiv:1701.00160 (2016).
- [37] Ramesh, Aditya, et al. "Zero-shot text-to-image generation." arXiv preprint arXiv:2102.12092 (2021).

[38] Floridi, L., Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds & Machines* 30, 681–694 (2020). <https://doi.org/10.1007/s11023-020-09548-1>

[39] Fiebrink, Rebecca. (2019). Machine Learning Education for Artists, Musicians, and Other Creative Practitioners. *ACM Transactions on Computing Education*. 19. 1-32. 10.1145/3294008.

[40]

<https://www.theatlantic.com/technology/archive/2019/03/ai-created-art-invades-chelsea-gallery-scene/584134/>

(accessed on Aug 5th)

[41] Bsteh, Sheila, and Prof Dr Filip Vermeulen. "From Painting to Pixel: Understanding NFT artworks." (2021).

[42] <https://quasimondo.com/> - AI Artist Mario Klingemann

(accessed on Aug 10th, 2021)

[43] <https://musically.com/>

Music Ally - a global music technology consulting group (accessed on Aug 15th, 2021)

[44] Carr, C. J., and Zack Zukowski. "Generating albums with samplernn to imitate metal, rock, and punk bands." arXiv preprint arXiv:1811.06633 (2018).

[45] <https://decrypt.co/72120/holly-herndon-nft>

Holly Herndon's DAO and NFT for her voice model (accessed on Aug 15th, 2021)

[46] [https://en.wikipedia.org/wiki/Holly\\_Herndon](https://en.wikipedia.org/wiki/Holly_Herndon) AI Artist Holly Herndon

(accessed on Aug 10th, 2021)

[47] Ames, C. (1989). The Markov process as a compositional model: A survey and tutorial. *Leonardo*, 22 (2), 175–187

[48] Davismoon, S., & Eccles, J. (2010). Combining musical constraints with Markov transition probabilities to improve the generation of creative musical structures. In *Proceedings of the European Conference on the Applications of Evolutionary Computation*.

[49] Wooller, R., & Brown, A. R. (2005). Investigating morphing algorithms for generative music. In *Proceedings of the International Conference on Generative Systems in the Electronic Arts*

- [50] Pachet, F., Roy, P., & Barbieri, G. (2011). Finite-length Markov processes with constraints. In Proceedings of the International Joint Conference on Artificial Intelligence.
- [51] Pachet, Francois. "The continuator: Musical interaction with style." Journal of New Music Research 32.3 (2003): 333-341.
- [52] Roads, C. (1979). Grammars as representations for music. Computer Music Journal, 3 (1), 48–55.
- [53] D. Cope. Experiments in Music Intelligence. A-R Editions, Madison, WI, 1996.
- [54] Biles, John. "GenJam: A genetic algorithm for generating jazz solos." ICMC. Vol. 94. 1994.
- [55] WASCHKA II, R. O. D. N. E. Y. "Composing with genetic algorithms: GenDash." Evolutionary Computer Music. Springer, London, 2007. 117-136.
- [56] Horowitz, D., Generating Rhythms with Genetic Algorithms, International Computer Music Conference, 1994, pp. 142-143.
- [57]  
<https://web.archive.org/web/20070205081404/http://evonet.lri.fr/eurogp2004/songcontes.html>  
 (archived from the web)
- [58] Santos, A., Arcay, B., Dorado, J., Romero, J. J., & Rodríguez, J. A. (2000). Evolutionary computation systems for musical composition. In Proceedings of the International Conference Acoustic and Music: Theory and Applications.
- [59] Ebcioğlu, Kemal. "An expert system for harmonizing four-part chorales." Computer Music Journal 12.3 (1988): 43-51.
- [60] P. M. Todd and G. Loy, editors. Music and Connectionism. MIT Press, 1991.
- [61] P. Toiviainen. Modeling the target-note technique of bebop-style jazz improvisation: An artificial neural network approach. Music Perception, 12 (4):399–413, 1995.
- [62] P. Toiviainen. Symbolic AI Versus Connectionism in Music Research. In E. Miranda, editor, Readings in Music and Artificial Intelligence. Gordon and Breach, 1999



[63] Rebecca Fiebrink and Baptiste Caramiaux. The machine learning algorithm as a creative musical tool, November 2016. arXiv:1611.00379v1.

[64] Bertin-Mahieux, Thierry, et al. "The million song dataset." (2011): 591-596.

[65] Porter, Alastair, et al "Acousticbrainz: a community platform for gathering music information obtained from audio." Müller M, Wiering F, editors. ISMIR 2015. 16th International Society for Music Information Retrieval Conference; 2015 Oct 26-30; Málaga, Spain. Canada: ISMIR; 2015.. International Society for Music Information Retrieval (ISMIR), 2015.

[66] Defferrard, Michaël, et al. "Fma: A dataset for music analysis." arXiv preprint arXiv:1612.01840 (2016).

[67] Fonseca, Eduardo, et al. "Freesound datasets: a platform for the creation of open audio datasets." Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.. International Society for Music Information Retrieval (ISMIR), 2017.

[68] Huang, Cheng-Zhi Anna, et al. "AI song contest: Human-AI co-creation in songwriting." arXiv preprint arXiv:2010.05388 (2020).

[69] Fiebrink, Rebecca, et al. "Toward understanding human-computer interaction in composing the instrument." ICMC. 2010.

[70] Fiebrink, Rebecca, and Perry R. Cook. "The Wekinator: a system for real-time, interactive machine learning in music." Proceedings of The Eleventh International Society for Music Information Retrieval Conference (ISMIR 2010)(Utrecht). Vol. 3. 2010.

[71] Bill Buxton. Sketching user experiences: getting the design right and the right design. Morgan kaufmann, 2010

[72] Ressel, Jean-Claude. "Computer Music Experiments 1964-..." Computer Music Journal (1985): 11-18.

[73] Baker, Janet M. "Using speech recognition for dictation and other large vocabulary applications." Applications of Speech Technology. 1993.

[74] Farooq, Umer, et al. "Human computer integration versus powerful tools." Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems. 2017.

[75] Ghias, Asif, et al. "Query by humming: Musical information retrieval in an audio database." Proceedings of the third ACM international conference on Multimedia. 1995.

[76] Cowen, Alan S., et al. "Mapping 24 emotions conveyed by brief human vocalization." American Psychologist 74.6 (2019): 698.

[77] Weidman, Amanda. "Sound and the city: mimicry and media in South India." Journal of Linguistic Anthropology 20.2 (2010): 294-313.

[78] - Bruscia, K. Improvisational Models of Music Therapy. Charles C. Thomas Publisher. Illinois, U.S., 1987.

[79] - Andersson, Anders-Petter, and Birgitta Cappelen. "Designing empowering vocal and tangible interaction." (2013).

[80] - Holbrow, Charles, Elena Jessop, and Rébecca Kleinberger. "Vocal vibrations." Proceedings of NIME. 2014.

[81] <https://www.humtap.com>  
(accessed on 10th Aug, 2021)

[82] <https://vochlea.com>  
(accessed on 10th Aug, 2021)

[83] <https://www.generativemusic.com> - Mobile apps by Brian Eno and Peter Chilvers  
(accessed on 10th Aug, 2021)

[84] Dhariwal, Prafulla, et al. "Jukebox: A generative model for music." arXiv preprint arXiv:2005.00341 (2020).

[85] <https://www.musictech.net/news/auxuman-release-ai-album/> - Humanoid AI musician  
(accessed on 13th Aug, 2021)

[86] Compton, Katherine. Casual creators: Defining a genre of autotelic creativity support systems. University of California, Santa Cruz, 2019. Chapter 13: Slow Creators

- [87] Russell, Stuart, and Peter Norvig. "Artificial intelligence: a modern approach." (2002).
- [88] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
- [89] Boulanger-Lewandowski, Nicolas, Yoshua Bengio, and Pascal Vincent. "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription." arXiv preprint arXiv:1206.6392 (2012).
- [90]  
<https://www.technologyreview.com/2020/07/20/1005454/openai-machine-learning-language-generator-gpt-3-nlp/>  
(accessed on 10th Aug, 2021)
- [91] Huang, Cheng-Zhi Anna, et al. "Music transformer." arXiv preprint arXiv:1809.04281 (2018).
- [92] Engel, Jesse, et al. "DDSP: Differentiable digital signal processing." arXiv preprint arXiv:2001.04643 (2020).
- [93] Hawthorne, Curtis, et al. "Onsets and frames: Dual-objective piano transcription." arXiv preprint arXiv:1710.11153 (2017).
- [94] Bogdanov, Dmitry, et al. "Essentia: An audio analysis library for music information retrieval." Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.. International Society for Music Information Retrieval (ISMIR), 2013.
- [95] Gero JS, Kannengiesser U (2004) The situated function–behaviour–structure framework. Des Stud 25(4):373–391
- [96] Zhang W, Wang J (2016) Design theory and methodology for enterprise systems. Enterp Inf Syst 10(3):245–248.
- [97] Jordanous A (2012) A standardised procedure for evaluating creative systems: computational creativity evaluation based on what it is to be creative. Cognit Comput 4(3):246–279

- [98] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training gans. In: Advances in neural information processing systems (NIPS). Barcelona, Spain
- [99] Karras T, Aila T, Laine S, Lehtinen J (2017) Progressive growing of gans for improved quality, stability, and variation. In: International conference on learning representations (ICLR). Toulon, France
- [100] Zbikowski LM (2002) Conceptualizing music: cognitive structure, theory, and analysis. Oxford University Press, Oxford
- [101] Hale CL, Green SK (2009) Six key principles for music assessment. Music Educ J 95(4):27–31.
- [102] Asmus EP (1999) Music assessment concepts: a discussion of assessment concepts and models for student assessment introduces this special focus issue. Music Educ J 86(2):19–24
- [103] Ellis, Daniel PW, et al. "The echo nest musical fingerprint." (2010).
- [104] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).
- [105] [https://soundcloud.com/openai\\_audio/sets](https://soundcloud.com/openai_audio/sets)  
(accessed on 13th Aug, 2021)
- [106] March, James G. "Exploration and exploitation in organizational learning." Organization science 2.1 (1991): 71-87.
- [107] City symphonies. Available from: <https://citysymphonies.media.mit.edu/>  
(accessed on 15th Aug, 2021)
- [108] van Troyer, A. (2017b). Score instruments: a new paradigm of musical instruments to guide musical wonderers. PhD thesis, Massachusetts Institute of Technology