NORTHWESTERN UNIVERSITY

The Neural Tape Loop: Controllable and Expressive Generative Modelling for the Sound Arts

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Hugo Flores García

EVANSTON, ILLINOIS

July 2025

Abstract

State-of-the-art generative audio models rely on text prompting mechanisms as a primary form of interaction with users. While text prompting can be a powerful supplement to more gestural interfaces, a sound is worth more than a thousand words: sonic structures like a syncopated rhythm or the timbral morphology of a moving texture are hard to describe in text. They can be more easily described through a sonic gesture. I describe two technical research works exploring generative audio modeling with gestural and interactive control mechanisms: VampNet (via masked acoustic token modeling) and Sketch2Sound (via fine-grained interpretable control signals). I introduce the neural tape loop: a co-creative generative musical meta-instrument for experimental music and sound art designed and developed using practice-based research methods. I propose new interactive sound manipulation techniques based on the affordances of masked acoustic token models, and illustrate the musical capabilities of these techniques through four original creative works (a composed improvisation, two fixed media electroacoustic pieces, and a multichannel interactive sound installation) made in collaboration with sound artists, composers, and instrumentalists. Finally, I reflect on how engaging in a mixed creative and technical research practice can be a catalyst for culturally situated and artist-centered innovation and advancement in generative musical instrument design.

Acknowledgement

I'd like to thank my advisor Bryan Pardo. Bryan has been a truly exceptional mentor, and has been a valuable guide at matters much bigger than the topics being discussed here. I'd really like to thank Bryan for always encouraging me to stay creative and musically engaged. Bryan encouraged me to keep playing guitar when I had almost stopped (towards the end of the pandemic) and did not hesitate to support me when I told him I wanted to engage in both technical and creative work with the neural networks I was making. I'd like to thank Bryan for believing in me and always putting the effort into creating the optimal space to support me and do my thing!

I'd like to thank my dissertation committee, Mike Horn, Darren Gergle, Prem Seetharaman, Anna Huang, and Andrew McPherson. I really appreciate the thoughtful discussions, feedback and mentorship that I've received from each and every one of you. I'd like to give a special thanks to Prem, with whom I've had countless brainstorming sessions, intellectual discussions and coding jams, many of which ended up sparking a new research project or superpowering an existing one. Prem has been there to help me since before the very first day of my PhD (I got my first set of furniture for my Evanston apartment from Prem, who was moving out of his!).

I'd like to thank all of the mentors I've had throughout my time spent doing research internships. A huge thanks to Rachel Bittner and Jan Van Balen, who were my mentors during my first ever internship (!), at Spotify. A huge thanks to my Adobe SODA mentors (and soon to be teammates!), Justin Salamon and Oriol Nieto. You folks (and Prem) have a fantastic lab energy and dynamic and I can't wait to be a long-term part of it! Thanks to Rithesh Kumar, to whom I owe lots of my understanding of transformer architectures and diffusion models.

Thanks to my labmates: Annie Chu, Patrick O'Reilly, Julia Barnett, Max Morrison, and Ethan Manilow. Thank you for your friendship and support as we all navigate the seemingly endless marathon of doom that is a doctoral degree. I have learned so much from you all and am so happy to have had the chance to hang out, learn, and collaborate with you.

Thank you so much to all of my musical friends and collaborators: Aldo, Michael, Tori, Yvette,

3

TABLE OF CONTENTS

LIST OF FIGURES

9

CHAPTER 1

INTRODUCTION

1.1 Introduction

The tech industry today sees the expressive power of generative neural nets as a way to build a universal musical instrument: a system they claim can be used by anyone to make music, regardless of their musical background or level of expertise [1]. This line of thinking has impacted considerably upon the design and engineering choices made by scientists working on generative AI for music and sound design. Most of the work has focused on systems for "novice" or casual creators, primarily operated using a text-prompted interface.

While text prompting can be a powerful supplement to more gestural interfaces, an interface based *solely* on text prompts constrains the space of compositional decisions available to an artist to a fixed lattice dictated by what is possible to describe affordably in text. **A sound is worth more than a thousand words**: musical structures like a syncopated rhythm or the timbral morphology of a moving texture are hard to describe in text. They can be more easily described through a sonic gesture [2, 3].

For instance, a form of sonic guidance (like a vocal imitation) can effortlessly convey the ineffable temporal specificities, contours, and inflections of sound. *The human voice is a gestural sonic instrument* [4]: it allows us to realize sounds without having to perform any symbolic abstraction (i.e., putting a sound into words) beforehand. When humans communicate audio concepts to other people (rather than software), they typically combine descriptive language and vocal imitation [5, 6, 7]. In doing so, one approximates the audio by mapping the pitch, timbre, and temporal properties of the sound to those of the voice. This is a more natural method than describing the evolution of pitch, timing, and timbre via pure text descriptions [5] and recent work has shown its utility for query-by-example search of audio databases [8, 9]. Our voice is built into our bodies, and

advanced vocal techniques (e.g., speaking, singing) are developed continually in life, making the voice a promising interface for expressively creating sounds with generative models – one with a low floor (it is easy for a novice to get started), high ceiling (with work, virtuosity is possible), and wide walls (a wide range of possible outcomes at any skill level) [10]. While the voice is a powerful device accessible to most musicians and non-musicians, sonic prompts for a generative model can also be guided by other means of sound production: a person may clap, tap a drum beat on a desk, manipulate physical sound objects (e.g. jingling keys), or play a musical instrument. For instrumentalists, the space of possible sonic gestures to guide a generative model is further expanded, as the instrumentalist has access to the sonic affordances offered by their instrument. Through their instruments, musicians can convey controlled and specific sonic gestures, which can be used as input material to have a controllable and expressive interaction with a generative model.

If sonic/vocal guidance is a promising mode of interaction for a generative model, what kinds of sounds should our model make in response to our sonic gestures? In a sound artist's creative practice, the raw sound materials used in composition or improvisation (e.g. field recordings, instrument 'one-shots', found sounds, etc.) can be as personal and fundamental to the artist's voice and creative intent as the skills and techniques the artist chooses to employ. It would thus be undesirable for sound artists to be limited to using the sounds of a generic pretrained model trained on "stock sounds" or all-encompassing sounds as material. If every artist used the same generative model checkpoint to create sonic artworks, all of their creative outputs would contain traces of that checkpoint's aesthetic qualities. This is undesirable for sound artists, who likely desire to convey their own stories and styles in their works by using and manipulating their own found and created sound materials. Esling and Devis [11] make the analogy that the standardization of generative music as creative practice would resemble a "genetic drift" in creativity, negatively impacting the diversity and variance of creative practices and artifacts. Instead, sound artists should be able to provide their own **sound palette**: a small to medium-sized collection (ranging from a couple of minutes to a couple of hours) of sound recordings that we can use to fine-tune the "base" generative model; any generated outputs following this fine-tuning process would match the sonic qualities

of the provided sound palette.

My goal is to build artist-centered human-AI co-creation systems for creative expression in the sound arts. To do so, I had to advance the state of the art in areas relating to interactive music generation, controllable and interpretable audio generation, and human-computer interaction for human-AI co-creation interface design. In the work for this dissertation, I developed a set of machine learning systems that (1) enable **more gestural and controllable interactions than text-based generative models** , (2) support a **wider sonic palette** and **longer-term structure** than a realtime sound generation systems (like RAVE [12]) while still being (3) **fast enough to run interactively** and able to (4) **fine-tune to an end-user's sound palette**

An important question to ask is: who is this instrument made for? Setting out to build a universal musical instrument is an ill-formed goal; music is not a homogeneous blob, but an umbrella term encompassing countless evolving communities of artistic practice, each with a unique set of styles, techniques, and aesthetic values[13]. Ultimately, there is no universal musical instrument, but rather an ecosystem of instruments, all part of a dynamic ecology of affordances situated in their respective communities of practice [14]. The technical contributions in this dissertation were constantly informed by and fed back into my own background and creative practice: I am a computer musician, guitarist, and improviser, with a background in the traditions of Latin American, Jazz, Experimental and Improvised Music. Each of the technical contributions in this dissertation was followed by a period of creative practice to explore the creative possibilities of the current system as-it-was, with the objective of reflecting on the possible next steps, prioritizing the technical challenges that were more likely to lead to an improved long-term creative interaction. This process constituted a form of practice-based research [15], where each technical advancement was followed by a period of creative practice, exploring the creative affordances of the system in its current state and figuring out a new technical research direction to further advance my goal of building a generative musical instrument for creative sonic expression.

In addition to my technical contributions in the field of generative modeling for audio, this dissertation introduces a set of original creative works, including generative sound installations, fixed

media musical pieces, and live improvisations with a generative model. These uncover and draw on the musical affordances and interactions that could only be made possible by the technical contributions described above. By reflecting on my compositions, performances, and collaborations with other artists, I show how engaging in creative practice led to the discovery of new musical affordances implicit in a generative model, reusable design guidelines for building co-creative musicmaking systems, and the development of follow-up technical innovations.

1.2   Contributions

- **Contributions to interactive music generation (Chapter 2)** The first audio-conditioned latent generative music model fast enough for interactive (non-realtime) generation. With it, I designed prompting strategies to perform tasks that previous methods (autoregressive) were not designed to do: sound inpainting, extreme (data) compression, and creating rhythmic and timbral variations of an input sound recording by leveraging beat and onset tracking methods, making it an expressive and interactive looping and sampling tool. This work was published at the ISMIR 2023 conference under the name VampNet [16].

- **Contributions to controllable and interpretable audio generation (Chapter 3)** a controllable (interactive, non-realtime) sound generation system capable of generating sounds from interpretable, fine-grained time-varying control signals that can be easily extracted from any audio signal (e.g. loudness envelope, pitch contour, brightness), as well as text prompts. By leveraging these control signals as conditioning, this system is able to generate arbitrary sounds from sonic imitations and sketchlike control curves, like low-frequency oscillators (LFOs). This work was published the ICASSP 2025 conference under the name Sketch2Sound [17].

- **Contributions to human-AI co-creation for music (Chapter 4)**
  - a generative musical meta-instrument designed via practice-based research methods, leveraging one of the generative sound systems mentioned above to allow an artist to create musically meaningful timbral, rhythmic, and structural transformations of a

recorded loop.

- a practice-based research account discussing the design of the proposed meta-instrument, introducing a new set of musical techniques based on the affordances of masked acoustic token models. Through the discussion of previous performances, sound installations, and artist collaborations, I propose reusable design and compositional guidelines for building and playing with generative AI music co-creation systems.

1.3 Broader Impact

Mainstream music AI products (e.g. Suno [1], Udio [2], Google MusicFX DJ [3], Beatoven [4]) favor consumer-centric casual creation interfaces that promise accessible "anyone-can-make-music" musicmaking by facilitating long-form compositions with exclusively high-level interactions (e.g. a high-level text prompt like "*sunny, dreamy pop-funk-orchestral-gregorian chant love song*").

Instead, I aim to contribute to the discourse of a community of artists building their own generative models and instruments for musicmaking [18, 19, 20, 21]. Much like the lineage of non-generative digital musical instruments before them, many of these instruments are motivated by the artist/researcher's own artistic practice .

The design choices I've made for my proposed musical instrument reflect a more grounded, artist-centric interface, offering "lower-level" gestural control by letting users manipulate the momentary acoustic features of a sound produced by a generative model of a sound palette. For example, the proposed time-varying control signals would allow an artist to control the *immediate* (i.e., ten-millisecond-level) energy and brightness of a generated stream of rain sounds from field recordings recorded by the artist.

By giving artists the ability to fine-tune their own model on a sound palette, I am giving the artist the power to participate in the modelmaking process. Pioneering AI artists Holly Herndon and Mat Dryhurst [22] assert that "the AI model is the artwork; the data is the artwork; the protocol

---

[1] `suno.ai`
[2] `https://www.udio.com/`
[3] `https://aitestkitchen.withgoogle.com/tools/music-fx-dj`
[4] `https://www.beatoven.ai/`

coordinating it all is the artwork", and so giving an artist the ability to choose their own sound palette further asserts the artists ownership over the artwork being created.

My technical contributions, which make state-of-the-art generative audio models faster and more controllable, will also have an impact in other creative industries in the sound arts, like Foley sound design. Foley sound is a skilled and gestural performance art: performing a sound scene with sound-making objects and instruments (instead of arranging pre-recorded samples post hoc) allows sound artists to create fluent and temporally aligned sounds with a gestural touch [17]. Giving a Foley artist the opportunity to control a pre-recorded sound library with vocal gestures and text references could foster new techniques for Foley sound, especially in cases where a Foley stage may not be available, and arranging sounds from a pre-recorded library may be the only choice.

My work will be of interest to researchers in generative modelling for audio (i.e. speech, music, sound fx), human-AI co-creative interfaces and digital musical instruments. The contributions proposed advance the state-of-the-art in generative modelling, making the existing generation of two-stage generative sound models faster and more controllable, opening up new meaningful ways to interact with powerful sound synthesis systems. From a human-computer interaction standpoint, this work will help obtain a better understanding of the role generative models play in the context of music performance, as well as help machine learning researchers learn how to design generative music models that create artist-centric experiences.

1.4   Background

This section provides an overview of related state-of-the-art research in the academic fields exploring human-AI co-creation interfaces and systems, digital musical instrument design, and controllable sound generation techniques with generative deep learning. This grounds how this dissertation advances each field of study.

1.4.1 Human-AI Interfaces for Music Creation

Recent research studying human-computer interaction (HCI) for human-AI music co-creation highlights that current human-AI interface layers for deep generative models fall short when it comes to giving performers a sense of agency, control, and authorship [23, 24]. To date, most of the work addressing these issues has focused on building systems targeted towards casual or novice music creators [24, 25]. Few studies involve practicing musicians [26], often only at the end of a study, for evaluative purposes.

This is because deep learning is often presented as a technology that can *democratize music*[5][6][7] [27] – that is, it allows non-experts to engage in music-making and create sophisticated musical products. Large tech corporations and startups (who are the primary drivers of new AI music co-creation tools) aim to make "democratized" technology due to it's massively scalable business model [28]. Morreale et al. [29] notes that we should be cautious of these calls for *democratization*: the main problem why a person who wants to make music is not able to is not because our current instrument technology is inefficient, or that instruments take a long time to master. Rather, there are deeper intersecting reasons as to why someone may want to engage in music but is unable to, such as "exclusion because of socioeconomic resources, underfunded arts in schools, time poverty driven by capitalist forces, and exclusion from music-making communities because of gender, disability, body size, musical tastes... just to name a few" [29]. Sturm et al. note that "democratized" AI-music creation services are "difficult to distinguish from a form of technologically-mediated deskilling" [28, 30], as they reduce users' needs for musical competence, expertise, and or/education, all of which are recognized barriers to access in musicmaking. McPherson et al. [13] argue that making an instrument for "anyone to make music" is a rather misconstrued goal, as "music is not one homogeneous entity but rather an umbrella term encompassing a huge variety of genres, styles, and techniques.".

Ultimately, there is no ideal instrument that will be right for everyone, but rather an ideal

---

[5]https://www.rollingstone.com/music/music-features/suno-ai-chatgpt-for-music-1234982307/

[6]https://www.theguardian.com/technology/2024/apr/13/ai-generated-music-app-suno-ai-impac

[7]https://www.forbes.com/sites/davidhenkin/2023/12/05/orchestrating-the-future-ai-in-the-

ecosystem of many musical instruments, each of them with a different set of affordances, occupying their own acoustic niche [31] designed to serve the artistic needs of a particular aesthetic and sociocultural context. *So what should guide our design and technical goals when building interactive generative musicmaking systems?*

1.4.2 Digital Musical Instrument Design

As designers of a new generation of co-creative AI musical instruments, our design principles can be helpfully guided by the conceptual frameworks discussing the nature of musical instruments and their contexts [32].

Rodger et al.[14] warn us that traditional HCI design methodologies aren't a perfect fit when thinking of musical instrument design. Instead, they suggest us not to think of musical instruments as "discrete, self-subsisting objects" or physical/software devices for making sound, but rather as part of a dynamic ecology of affordances situated in their respective cultural, historical and conceptual contexts [33, 32]. Thus, any design studies carried out for an instrument will be most successful if they are to be situated within a cultural and artistic/stylistic context throughout the design process [34, 35, 36].

The past half-century has seen the emergence and growth of a community of digital musical instrument (DMI) makers and performers, leading to the rise of academic communities focused on Computer Music [37] in the 1970s and New Interfaces for Musical Expression (NIME) in the early 2000s [38]. This community of practice stands out, as they have nurtured a relatively new tradition that entangles instrument-making and creative musical practice together [39]. Contemporary interfaces for engaging in the practice of musicking [40] have both challenged the notion of what a musical instrument *is* [33, 41] and blurred the line between composition and performance [42].

While working on a digital musical instrument, a DMI maker may play the roles of instrument-builder, composer and performer all together [39], engaging in an iterative loop of *design time* and *play time* [43], where each play session informs what the direction should be for the next round of design. This process encourages the instrument builder to stop and play with an instrument as-is to

evaluate its current functions and aesthetic qualities before committing to another round of design and building.

Engaging in this iterative design-play process lends itself well to practice-based research methods [44, 45, 46]. Practice-based research methods stem from research in the arts and humanities, aiming to capture the complex, non-linear processes inherent to artistic practice [47].

Since before the establishment of the NIME conference itself, NIME research practices in NIMEs covered a wide variety of disciplines [48], from technical reports and system designs to musicological, historical, and critical perspectives on new musical instruments and the music made with them. However, Gurevich [48] noticed that in the 2010s, NIME papers skewed heavily more toward technical reports and scientific contributions, highlighting the implicit increasing pressure for every NIME publication to contain quantifiable outcomes and "scientific contributions".

Being able to offer a diversity of research perspectives is arguably a potential strength of the NIME community, and Gurevich calls for people engaged in practice-based research (PBR) to "examine its goals, expectations, and parameters with the aim of clarifying what could constitute legitimacy within the PBR community". A recent publication by Pelinski, McPherson and Fiebrink calls for more Technical Practice Research [45] in the field of NIME, arguing that there is a lot of knowledge to be gained from the "messy, non-linear unfoldings, reflexive discomfort and nuance" [49, 45] of practice as opposed to the linear design narratives often conveyed in traditional scientific research.

In Chapter 4, I further discuss the role of musical practice in musical instrument design research. Chapter 5 is a practice-based research account of my design of a generative musical meta-instrument, along with several creative works to accompany it. In Chapter 5, I propose a set of techniques for manipulating acoustic tokens to achieve different musical outcomes, as well as propose design guidelines for building and playing generative AI music co-creation systems for practicing musicians.

two-stage audio generation

stage 1

(encoder) (decoder)

audio latents
(10 - 80 Hz)

input audio

time

output audio

VAE / RVQ-VAE

stage 2

input controls

audio latents

text prompt

"bees buzzing"

control signals

language model
//
diffusion model

audio latents
(generated)
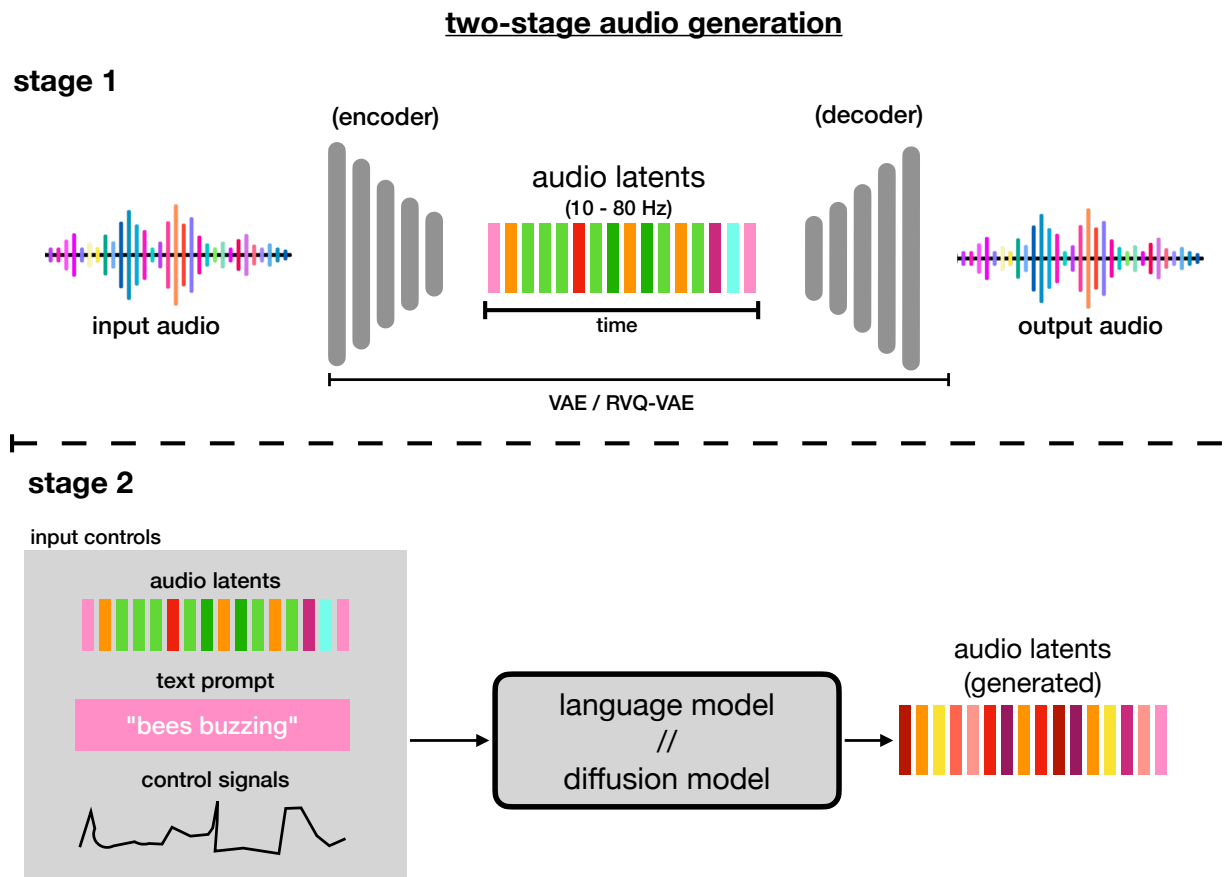
Figure 1.1: Most state-of-the-art audio generation systems employ a two-stage approach: An encoder-decoder learns to produce a compressed representation of an audio signal (the audio latents) at a low signal rate (10-80Hz) and reproduce audio signals from this compressed representation. Separately, a generative model learns to create new sequences of audio latents conditioned on controls like text prompts. The generated latents are then decoded into new audio signals by the decoder.

1.4.3   Controllable audio generation

The state-of-the-art approach for offline (non-realtime) generation of sounds (as well as other signals, like images) with long-term structure (i.e., in the order of 10 to 90 seconds) is a two-stage approach [50]. Originally conceived in the image domain, two-stage generation models [51, 52] led audio researchers to explore text-to-audio generation. Text-conditioned sound generation models like AudioLDM [53], AudioLM [54] and MusicLM[54, 55], MusicGen[56], MAGNet [57] all follow a similar two-stage approach to audio generation (see Figure 1.1):

- (1) In the first step, we train an **encoder/decoder** to compress raw sound waveforms into

sequences of latents (or "tokens" in the discrete case [55]) with a slower time resolution than audio samples (e.g., 40-90Hz [58] vs. 44.1KHz for raw full-band audio). In the continuous case, the autoencoder is typically a variational autoencoder (VAE) [59]. In the discrete case, a residual vector-quantized VAE (RVQ-VAE[60]) is used instead.

- (2) In the second step, we train a **prior** model to generate sequences of latents. These latents are easier to model than long sequences of raw audio samples due to their lower temporal resolution, allowing us to generate new sequences of audio latents using diffusion [53, 61, 62, 63] or language modeling [55, 54, 64, 57], and thus generating novel audio waveforms with "long"-term structure, usually in the range of 30 seconds [56] to 3 minutes [65].

- At inference time, a user provides their contol input (e.g. a text prompt, sound prompt, or control signal) to the prior model, which generates a new sequence of audio latents (or tokens). These are then passed to the decoder from step 1, which converts the generated output into audio.

1.4.4 Autoregressive Language Modeling

In the early 2020s, two-stage generative music models used approaches based on autoregressive language modeling [55]. Autoregressive sampling is slow in nature due to the high number of steps required at inference time (one per acoustic token). For large model sizes, this limits a model's ability to function interactively, as they are much slower than real time and may have long waiting times during processing.

In Chapter 2, I will discuss VampNet [16], the model I developed which was the first[8] system that used masked token modeling methods [66] for residual vector-quantized acoustic tokens, reducing the number of sampling steps required to generate audio by an order of magnitude compared to an autoregressive language model.

Further, autoregressive models inherently restrict downstream applications for interactive music editing, as each generated token is only conditioned on the previous tokens. For an autoregres-

---

[8]Note: Google's Soundstorm [60] was concurrent work that also took a similar approach

sive model to perform tasks like inpainting ("filling in the middle"), one must rearrange the data during training [67]. On the other hand, VampNet's bidirectional attention allows it to perform tasks like inpainting, music compression, and creating variations and timbre transfers of a sound prompt. Notably, VampNet is capable of generating sounds that follow an input sonic gesture's overall structure while retaining the timbral qualities of the model's fine-tuning data.

Although more recent methods optimize the speed of audio latent diffusion models for *high throughput* [68] (e.g., generating 90s of audio in 200ms), these models are mostly still non-causal, meaning that they aren't designed for low-latency realtime audio streaming, with the recent exception of [69].

That being said, VampNet's fast sampling method made it possible to run it interactively in a loop, making it suitable for instrumental interaction in a live looping setting. **We can leverage large acoustic latent models in a live looping scenario**: after a loop has been recorded into a buffer, this buffer can be sent out for processing with the deep learning model. The recorded audio clip can keep looping in the looper interface while the model is processing. Once processing has finished, we can replace the contents of the original loop with our new "generated" loop. Using this approach, a musician can interactively co-create with a generative model by adding layers of sound to the loop along with the generative model, in a back-and-forth interaction where musical ideas are iterated on both by the performer and the generative model during a live performance. To demonstrate this interaction concept, I built a live looping instrument called unloop [9].

1.4.5 Latent Diffusion in Audio Generation

In addition to masked acoustic token modeling approaches [16, 64, 57, 70], latent audio diffusion [53, 62] models have been proposed as another faster alternative to autoregressive acoustic token models. They are now a common approach to two-stage audio generation [71, 72, 68, 65, 69, 73, 61, 63, 68]. Instead of modeling a multi-level sequence of discrete acoustic tokens produced by a residual vector quantized variational autoencoder(RVQ-VAE), latent diffusion methods instead

---

[9]https://github.com/hugofloresgarcia/unloop

learn to model sequences of the continuous latents of a variational autoencoder (VAE), removing some of the added complexity introduced by residual vector-quantized acoustic tokens while still being able to synthesize acoustic latents in fewer sampling steps than an autoregressive model.

Although masked acoustic token modeling and audio latent diffusion models may appear to be wildly different approaches, several studies have pointed at the relationship between these two approaches [74], noting that masked token models can be framed as a special case of discrete diffusion, and that diffusion models perform autoregression in the frequency domain [50].

1.4.6 Controllability Beyond Text

Because two-stage generative audio models often aim to model a wide multi-timbral distribution of sounds (e.g. various musical styles and instrumentations or a large library of foley sound effects), they require auxiliary conditioning to generate coherent samples in a controllable manner. The most common way [55, 56, 57, 53, 65, 62, 75, 76, 77, 78, 79, 80, 81] to condition these models is through text conditioning. While text conditioning is a good way to provide high-level guidance of the timbre or general structure of a generated sound, it does not allow for gestural, fine-grained temporal control over a generated sound, which is necessary in many scenarios in music performance, composition, and sound design.

On the other hand, text-to-sound models often suffer from limitations to their control as they only offer control over high-level aspects of the resulting signal, such as *what* sound categories are present in a recording, but not precisely *when* or *in what order* these sounds occur [82]. Furthermore, current paired (text, audio) datasets lack fine-grained textual descriptions of the timbral idiosyncrasies and spectromorphology [83] of a sound (e.g. 'It is a gentle swoosh that lasts 10 seconds and glides up from Middle C on the piano to the C one octave above it.'). This makes training a text-to-sound model to follow momentary instructions impractical (and the end result would be no different than using an existing programming language for sound synthesis like SuperCollider).

Overall, text-conditioning as a sole control method for a generative sound model faces problems related not only to the data available but also due to the nature of text descriptions themselves.

24

It would be clunky to describe the desired precise rhythm, microtiming, spectral envelope, and resonant frequencies of a bell sound in a text description. Consider the following prompt: *a small bell with a strong partial at 600 Hz, playing a loose rhythm with a triplet feel, occasionally skipping the first beat, affected with a low pass filter that slowly makes the sound darker over each repeated triplet grouping. The filter ranges between a cutoff at 2kHz and 800Hz with high Q factor, giving the bell a 'wah'-like sound.* Even if the model could even handle such a complex prompt (state-of-the-art text-to-sound models couldn't), this sonic idea could be more easily communicated to the generative model by providing a vocal imitation that imitates a bell sound at the desired rhythm and with the desired temporal spectral qualities. An accompanying piece of conditioning (like a text prompt saying "bell" or an audio prompt with the desired bell sound) could complement the vocal imitation, giving the model a reference timbral template to use with the given vocal gesture.

To improve over a purely text-to-sound input paradigm, some works in the music domain (including VampNet) conditioned on masked audio tokens [16], multiple parallel instrument stems [70], melody and text [56], chord and melody [84] or multiple structural control signals like song structure and dynamics [72]. Since most of the systems above are designed for musical applications (i.e., they prioritize the role of pitch and harmony in the resulting music over the timbral morphology of a sound), these approaches are not capable of controlling the fine-grained temporal behavior of an arbitrary sound object, like the loudness, pitch contour and time-varying brightness of the sound of an engine starting up. This kind of gestural control over the spectromorphology of a sound is of importance to other sonic art disciplines like improvised music, experimental music, electroacoustic music, sound art, sound design, etc.

In the speech domain, Morrison et al. [85] propose a fully interpretable and disentangled representation for speech, which allows for fine-grained control over the pitch, loudness, and phonetic pronunciation of speech. A notable result of this representation is the ability to generate "onomatopeias" by encoding non-speech sounds into their proposed speech representation and decoding them with a speech generation model. One could imagine achieving the reverse: generating arbitrary sounds and textures from vocal imitations by leveraging an intermediate representation

where sounds and their vocal imitations are correlated in one way or another.

In Chapter 3, I will discuss Sketch2Sound: a sound generation method conditioned on a set of interpretable, time-varying control signals that can be suitable for generating variations of sound objects, editing existing sound objects by modifying their extracted control signals, as well as gesturally generating sound-objects via a (text-prompted, *optionally*) vocal imitation. With a vocal imitation, one can describe the spectromorphology of a target sound in an embodied manner, making it an attractive interaction paradigm for timbrally-focused music, sound design, and other sound art disciplines.

1.5   Digital Musical Instruments and Generative Modelling

Before two-stage audio generation approaches became common practice, lightweight single-stage audio generation methods were built for use in realtime scenarios. One of these techniques is differentiable digital signal processing (DDSP) [86, 87]. The original DDSP model [86] extracts the pitch and loudness contour from an input source signal and uses it as conditioning for a differentiable decoder model with built-in DSP modules, like a harmonic plus noise synthesizer. DDSP quickly gained attention as a timbre-transfer tool [88]. However, DDSP's reliance on the harmonic plus noise synthesizer as a "sound decoder" meant that the model can only synthesize single monophonic instruments, like "violin" and "flute". Several other works improved over this limitation by adding different synthesis modules [87, 89]. Instead of using DDSP's harmonic plus noise synthesizer as a way to decode sounds from control signals, the work proposed here leverages the decoder of a neural autoencoder, like DAC [58] or RAVE [12], to map from a compressed latent representation to audio samples, allowing us to synthesize more than just monophonic sources, expanding the model's capabilities to polyphonic sources as well as complex, noisy sound textures.

Another notable example of a recent realtime audio synthesis model is RAVE. Introduced by Caillon et al. at IRCAM, RAVE [12] is a realtime variational autoencoder for audio that was introduced and quickly became adopted by a community of experimental musicians and music technologists [18, 90], while also becoming the flagship timbre transfer model architecture for

realtime AI-powered audio plugin company Neutone[10]. Although a VAE model (like RAVE) is an "autoencoder" model, it is constantly subject to creative misuse by a community of computer musicians. For example, a RAVE model can be used for timbre transfer by first training on a target distribution of sounds (e.g. darbouka sounds) and using a different distribution of sounds as input (e.g. the voice). In other cases, the RAVE latent space can be manually manipulated by mapping each latent in the RAVE model. While these techniques produce audio signals that would be deemed "unrealistic" and "low-quality" from an engineering perspective, the model's (relative) immediacy is preferable to a clunky two-stage generative model in a situation where being able to achieve gestural and highly interactive realtime interactions, like in a musical instrument [19].

For example, the RAVE [12] model is a popular generative model choice for instrument makers, thanks to its realtime inference speed and its integration into Max/MSP and other creative coding languages and environments. Moisés Horta Valenzuela built semilla.ai [18], a musical instrument that connects RAVE latent spaces to ancient Mesoamerican divination through "maíz throwing" technique. Pelinski et al. [91] built a pipeline for recording datasets and training neural networks like RAVE on these datasets using the Bela platform. Shepardson and Magnusson introduced the Living Looper [90], a live looper that records RAVE latent vectors to create "living" versions of the guitar loops with the hope of creating a co-creative instrument with agential behavior. Visi's Sophtar[21] is a tabletop string instrument that incorporates self-playing modes involving feedback and RAVE model processing. Privato et al. built Stacco [19], a musical instrument that leverages magnetic interactions to drive RAVE models. A follow-up physical interface to Devis et al.'s controllable RAVE model [92] was NeuroRack[11], which situated generative audio synthesizers in the context of modular synthesis, allowing for the model's control signals to be manipulated via control voltage (CV) signals.

One of the disadvantages of using the aforementioned single-stage approaches like RAVE is that, possibly due to their size, these models are trained on unitimbral sound distributions (e.g., just violin, just darbouka, just speech). A bidirectional, multi-distribution model (like VampNet or

---

[10]https://neutone.ai/
[11]https://github.com/acids-ircam/neurorack

Sketch2Sound) is more flexible and allows one to explore a wider space of timbres with a single neural network. This is a powerful advantage of using text-based conditioning as a *complement* (Chapter 3) to the time-varying control signals (e.g., when creating sounds via vocal imitation) as it allows a user to provide a strong conditioning signal that can reference a desired piece of sound material from the model's sound palette (i.e., it's training or fine-tuning data).

1.5.1   Sound palettes: generative models as corpus-based musical instruments

Sampling (the musicmaking technique where sounds are stored and replayed in musical arrangements, not to be confused with *sampling* from a probability distribution) has been a long-standing tradition in musicmaking: our "modern" understanding of sampling has its origins in musique concréte and tape music in the 1940s [93].

In a sound artist's creative practice, the sound materials (samples) used in a performance or composition can be an integral part of an artist's style and message. Sound artists can manipulate sound samples using sample-based instruments like samplers, concatenative synthesizers, granular synthesizers, etc. These sample-based instruments allow for gestural control of the provided sound sample: with a sampler, a sound artist can "shape" a sampled sound with a musical gesture and effectively employ it in a composition or improvisation.

On a similar vein, corpus-based instruments [94, 95, 96, 97] allow a performer to work with and manipulate large sound libraries, often by incorporating audio feature extraction techniques like Mel-frequency Cepstral Coefficients (MFCCs) to allow content-based search through these large collections of sound and query in realtime during a performance, or for offline composition work [98]. Many different interactions for corpus-based musicmaking systems have been explored. For example, one can interact with sound corpora through gestural sensor mappings [99], navigating 2-dimensional timbre spaces [100], or by providing a source audio signal for a concatenative synthesis algorithm to match and follow [101, 98].

One could think about a generative audio model as another form of a corpus-based instrument, like a **probabilistic sampler**: a system can create new sounds by modeling a probability

distribution of a large sound sample library. Furthermore, a sound artist may be able to "shape" sounds from said sound library by interacting through conditioning signals (loudness curves, text prompts). Given that the sound material can be as important as the system that synthesizes from it, a research effort must be made to empower sound artists with the ability to personalize a generative audio model to follow the artist's personal sound sample library, or **sound palette**.

Fine-tuning a generative model can be a powerful way to personalize it to conform to an artist's sound palette. Latent audio generative models could extend the practice of sampling further by allowing the sound artist to manipulate and reference a very large sound corpus with novel control paradigms like text and audio prompting (proposed in this dissertation).

LoRA (low-rank adaptation) [102] is a method for fine-tuning large transformer models efficiently by learning low-rank decomposition matrices with much fewer learnable parameters than those required to fine-tune the full transformer model. Through LoRA fine-tuning, VampNet can be fine-tuned with custom sound palettes (or collections of sounds to be used as material for a generative model) on a consumer GPU in under a day. To the best of my knowledge, VampNet was the first generative latent audio model that adopted LoRA fine-tuning to allow for a user to create their own generative model from their own sound palette. This allows sound artists to bring their own colors to a generative sound co-creation system, which is essential if our goal is to build artist-centered generative co-creation systems.

These methods are incorporated into the VampNet [16] system, which allows sound artists to fine-tune a VampNet model on a custom sound palette in under a day on a consumer GPU. As a result, the generative music systems proposed here give artists the power to use their sound material as fine-tuning data for the built-in generative sound model, allowing them to "shape" their sounds with the expressive power of modern generative modeling.

The following chapter will discuss VampNet, the first[12] masked acoustic token modeling system. VampNet was my first work within generative modeling for audio and has become a central part of my creative practice with generative models (Chapter 5).

---

[12]concurrent: Soundstorm [64]

VAMPNET: MASKED ACOUSTIC TOKEN MODELING
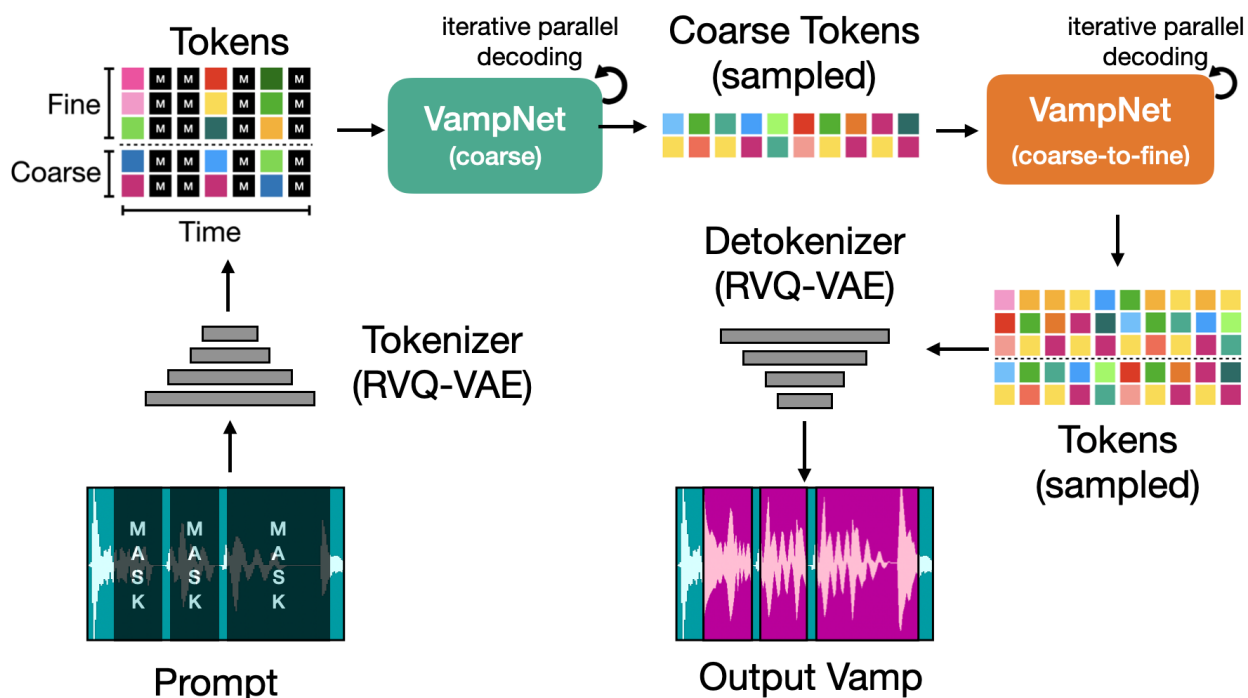
Figure 2.1: VampNet overview. We first convert audio into a sequence of discrete tokens using an audio tokenizer. Tokens are masked, and then passed to a masked generative model, which predicts values for masked tokens via an efficient iterative parallel decoding sampling procedure at two levels. We then decode the result back to audio.

In this Chapter, I discuss VampNet [16], a system for audio-prompted audio generation. This work was previously published at the ISMIR 2023 conference [16]. While the following work was led by me, it is important to note that this work was done in collaboration with Prem Seetharaman, Rithesh Kumar, and Bryan Pardo. Thus, uses of the pronoun "we" in the remainder of this chapter refer to work led by me, in collaboration with the researchers mentioned above.

In recent years, advances in discrete acoustic token modeling have resulted in significant leaps in autoregressive generation of speech [103, 104] and music [55]. Meanwhile, approaches that use non-autoregressive parallel iterative decoding have been developed for efficient image synthe-

sis [66, 105]. Parallel iterative decoding promises to allow faster inference than autoregressive methods and is more suited to tasks like infill, which require conditioning on both past and future sequence elements.

In this work, we combine parallel iterative decoding with acoustic token modeling, and apply them to music audio synthesis. To the best of our knowledge, ours is the first [1] extension of parallel iterative decoding to neural audio music generation. Our model, called VampNet, can be flexibly applied to a variety of applications via token-based prompting. We show that we can guide VampNet's generation with selectively masked music token sequences, asking it to fill in the blanks. The outputs of this procedure can range from a high-quality audio compression technique to variations on the original input music that match the original input music in terms of style, genre, beat and instrumentation, while varying specifics of timbre and rhythm.

Unlike auto-regressive music models [104, 55], which can only perform music continuations – using some prefix audio as a prompt, and having the model generate music that could plausibly come after it – our approach allows the prompts to be placed anywhere. We explore a variety of prompt designs, including periodic, compression, and musically informed ones (e.g. masking on the beat). We find that our model responds well to prompts to make loops and variations, thus the name VampNet [2]. We make our code open source[3] and highly encourage readers to listen to our audio samples[4].

2.1 Background

Two-stage approaches to generative modeling have gained traction in image [105, 66, 106, 107] and audio [104, 55, 64, 56] synthesis, largely in part due to their computational efficiency. In the first stage, a discrete vocabulary of "tokens" is learned for the domain of interest. The input is put through an encoder to obtain these tokens, which can be converted back into the input domain

---

[1]While our work was under peer review, Google released SoundStorm [64], which leverages a similar parallel iterative decoding approach to ours.

[2]To vamp is to repeat a short passage of music with variation.

[3]https://github.com/hugofloresgarcia/vampnet

[4]audio samples: https://tinyurl.com/bdfj7rdx

via a corresponding decoder. In the second stage, a model is trained to generate tokens, and is optionally given some conditioning (e.g. previous tokens, a text description, a class label) to guide generation.

2.1.1 Stage 1: Tokenization

In images, visual tokenization has been leveraged for state-of-the-art classification [108] and synthesis [66, 106, 109, 107]. The most popular approach is to use vector quantization on a latent space. Similar approaches have been explored for audio [110], but until recently such approaches have been restricted to low sampling rates (e.g. 16khz), or have been restricted to speech audio. The "sampling rate" of the latent space (the number of latent vectors required every second to represent audio) is a critical aspect of the tokenization scheme. The lower the sampling rate of the latent space, the easier the next stage (generation) will be to accomplish. Recently, methods based on residual vector quantization [111, 112] have been proposed for audio tokenization at high compression rates with good reconstruction quality of high-sample-rate audio.

The primary work we leverage for audio tokenization is the Descript Audio Codec (DAC) [58]. With DAC, audio is encoded into a sequence of tokens via a fully convolutional encoder. The output of this encoder is then quantized using a hierarchical sequence of vector-quantizers [109]. Each quantizer operates on the residual error of the quantizer before it. Because of this residual vector quantization, DAC is able to reconstruct audio with very high quality, at a high compression ratio. It, along with its predecessors [112, 111], are instrumental in enabling audio language models like AudioLM [104], MusicLM [55], and VALL-E [103]. While we later briefly describe our tokenizer, the key contributions of our work are applicable to the output of any audio tokenizer and our specific audio tokenizer is not the focus of this work.

2.1.2 Stage 2: Generation

Given audio encoded as tokens, one common approach is to use an autoregressive model [113] for generation. We will cover the other common approach (diffusion models) in Chapter 3. State-of-
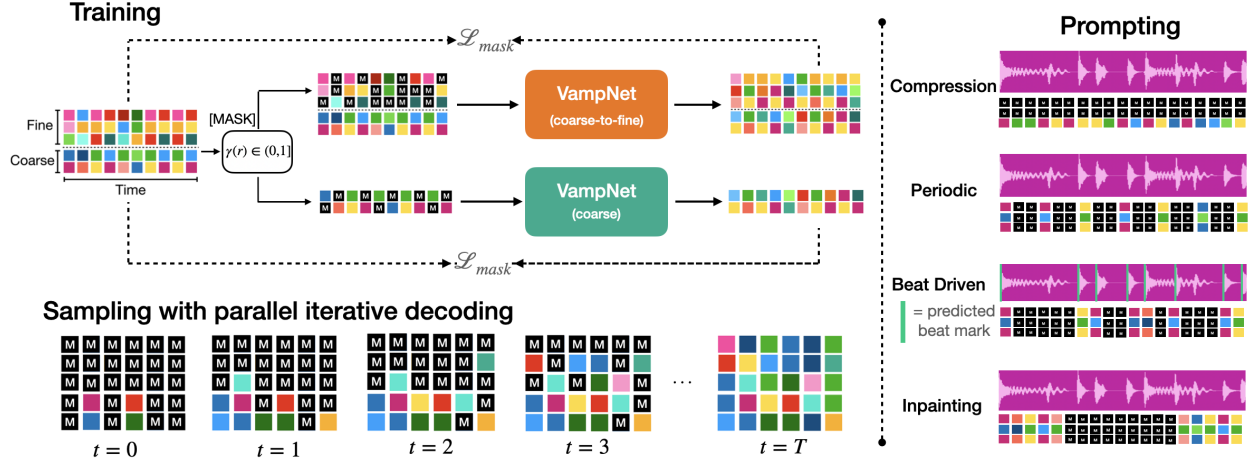
Figure 2.2: Training, sampling, and prompting VampNet. **Training**: we train VampNet using Masked Acoustic Token Modeling, where we randomly mask a portion of a set of input acoustic tokens and learn to predict the masked set of tokens, using a variable masking schedule. Coarse model training masks coarse tokens. Coarse-to-fine training only masks fine tokens. **Sampling**: we sample new sequences of acoustic tokens from VampNet using parallel iterative decoding, where we sample a subset of the most confident predicted tokens each iteration. **Prompting**: VampNet can be prompted in a number of ways to generate music. For example, it can be prompted periodically, where every Pth timestep in an input sequence is unmasked, or in a beat-driven fashion, where the timesteps around beat markings in a song are unmasked.

the-art (SOTA) audio generation approaches at the time we developed VampNet include AudioLM [104], MusicLM [55], and JukeBox [114]. These all use an autoregressive approach, generating each acoustic token in the sequence in a step-by-step fashion using transformer-based [115] decoder-only models. Autoregressive sampling is slow in nature due to the high number of steps required at inference time [66]. Further, autoregressive models inherently restrict downstream applications, as each generated token is only conditioned on the previous tokens. For an autoregressive model to perform tasks like inpainting ("filling in the middle"), one must re-arrange the data during training [67].

In language, masked modeling has been used extensively as a pre-training procedure for high-quality semantic representations [116]. This procedure has also been extended for representation learning in images [117] and audio [118]. Masked modeling for representation learning generally has a constant mask probability. For example, in BERT [116], tokens are masked 15% of the time during training. It has been shown that this approach is equivalent to a single-step discrete

diffusion model [74], that uses masking for its noising procedure. Therefore, we can extend masked modeling to masked generative modeling by varying the probability of masking a token during training. This was done for image generation in MaskGIT [66], and in language [74]. Similar to diffusion modeling [119, 120], which seeks to synthesize data starting from random noise through a series of denoising steps, masked generative modeling seeks to synthesize data starting from completely masked data through a series of "unmasking" steps.

Key to the efficiency of MaskGIT and related approaches is a *parallel iterative decoding procedure*. In parallel iterative decoding, the model predicts every token in the output sequence in a single forward pass. However, after just one forward pass of the model, the output often does not have high quality. The output of the first sampling step is re-masked, with a lower masking probability, and then put through the model again. In this way, masked generative models can efficiently refine their output, resulting in high quality generation.

In unconditional generation tasks, the model is asked to generate a realistic sample from the target data distribution from scratch, without any guidance. This is a difficult problem, as many target data distributions are highly multimodal. Unconditional generative models are susceptible to mode collapse [121], blurry samples, mode averaging, and other issues[122]. Therefore, some conditioning is helpful as it provides some signal for the model to resolve the multimodality. Conditioning is also a commonly used method to guide the output of the system towards desired content.

Conditioning can take the form of a class label, a genre tag or lyrics [114], or an associated text description [123, 107, 55]. Conditioning can also be applied at every timestep, like the semantic tokens of AudioLM [104], or aligned text or phonemes for text-to-speech generation [103].

In this work,we adopt a masked generative modeling approach with a parallel iterative decoding procedure, inspired by work in vision such as *MaskGIT* [66] and *Paella* [105], as illustrated in Figure 2.1. We do not apply any conditioning beyond that provided by the unmasked tokens in our encoded audio. As we show later, different approaches to masking, applied at inference time, can be used to steer generation in useful and artistic ways.

In training, tokens are masked randomly throughout the sequence. The model is then asked to predict the value of each of the masked tokens in a single forward pass, but it is conditioned on all of the unmasked tokens, both in the future as well as in the past. We vary the number of tokens that are masked during training, allowing us to generate audio at inference time through a sampling procedure. We now describe our method in more detail.

2.2 Method

We adapt the procedure of *Masked Visual Token Modeling*, proposed in MaskGIT [66] to audio, accounting for several key differences between the vision and audio domain. We call our approach *Masked Acoustic Token Modeling*.

2.2.1 Masked Acoustic Token Modeling

We first train an audio tokenizer based on the techniques used to develop the Descript Audio Codec (DAC) [58]. Unlike the visual tokens of MaskGIT, our acoustic tokens are hierarchical in nature due to residual vector quantization. As a first step, the audio signal x is encoded at each time step t as a a D dimensional latent vector Z. We then quantize Z using N vector quantizers. Quantizer 1 produces $\hat{Z}_1$, a quantized approximation of Z that has residual error $R_1 = Z - \hat{Z}_1$. Thereafter, the residual from each quantizer i is passed to the next quantizer $i + 1$, which produces a quantized approximation of the remaining residual error: $R_i \approx \hat{Z_{i+1}}$. Vector Z is reconstructed by summing the output of the N quantizers: $Z = \sum_{i=1}^{N} \hat{Z}_i$.

Since the encoded signal is represented as a quantized vector of N discrete tokens at each timestep, we have N tokens that can be masked or unmasked at each timestep. Rather than attempt to generate all tokens at once, we instead split the N tokens into N_c "coarse" tokens, and N_f "fine" tokens, as in AudioLM. We then train two generative models: one that generates the fine tokens given the coarse tokens as conditioning, and one that generates the coarse tokens given a sequence of coarse tokens. To generate a sample (Figure 2.1), we chain the two models together. First, we apply the coarse model to generate a sequence of coarse tokens. Then, we apply the coarse-to-fine

model to generate the fine tokens. We decode the tokens to a 44.1khz waveform using the decoder of our audio tokenizer.

2.2.2 Training procedure

Let $\mathbf{Y} \in \mathbb{R}^{T \times N}$ be a matrix representing the output of the encoder for some audio segment. Each element $y_{t,n}$ in $\mathbf{Y}$ is a token from the nth level codebook at timestep t. Let $\mathbf{Y}_M$ be the set of all masked tokens in $\mathbf{Y}$ and $\mathbf{Y}_U$ be the set of all unmasked tokens in $\mathbf{Y}$. The model generates a probability distribution over the set of possible codebook values for each token $y \in \mathbf{Y}_M$, given the unmasked tokens and the model parameters θ. The training objective is to maximize the probability of the true tokens. This corresponds to minimizing the negative log likelihood.

$$\mathcal{L} = - \sum_{\forall y \in \mathbf{Y}_M} \log p(y|\mathbf{Y}_U, \theta) \tag{2.1}$$

To predict the masked tokens, we use a multi-layer bidirectional transformer, which predicts the probabilities of each possible token at every timestep, for every quantizer. If each quantizer has a codebook size of C possible values, and there are N quantizers, then the last layer of the network will be a fully connected layer of shape (E, CN), where E is the dimensionality of the output of the last layer. We then reshape this output into (EN, C), and compute the cross-entropy loss between the ground-truth one-hot token and the predicted token. Because the transformer is bidirectional, it can attend to all tokens in the input sequence to optimize the loss for each token.

For the coarse-to-fine generative model, the input sequence always contains N_c coarse tokens, and the masking operation is restricted to the N_f fine tokens. The last layer of this network only predicts masked fine tokens. Otherwise, the training procedure for both models is identical.

2.2.3 Sampling

We follow the same iterative confidence-based sampling approach used in MaskGIT. More concretely, given Y_M as the set of masked tokens and Y_U as the set of unmasked tokens, do:

1. **Estimate.** For each masked token y in Y_M, estimate the conditional probability distribution over its vocabulary of codebook values V.

2. **Sample.** For each masked token, sample from the distribution to generate an associated token estimate $\hat{y} \in V$. We don't use any sampling tricks in this step, sampling from the categorical probability distribution for each token as-is.

3. **Rank by Confidence.** Compute a confidence measure for each of the sampled tokens by taking their prediction log-probabilities and adding temperature-annealed Gumbel noise to them:

$$confidence(\hat{y}_t) = log(p(\hat{y}_t)) + temp \cdot g_t \qquad (2.2)$$

where $\hat{y}_t$ is a token estimate at timestep t, g_t is an i.i.d sample drawn from Gumbel(0,1) [124], and $temp$ is a hyperparameter that is linearly annealed to 0 over the number of sampling iterations. Then, sort the set of sampled token estimates by the confidence computed above. We find that high temperature values (e.g. > 6.0) result in higher quality samples.

4. **Select.** Pick the number of tokens to mask at the next sampling iteration, k, according to the masking schedule [5]. Take the k lowest confidence estimates and toss them out, re-masking their tokens. Place the remaining high-confidence token estimates in Y_U, removing their tokens from Y_M.

5. **Repeat** Return to step 1 until the number of iterations has been reached.

2.2.4 Prompting

Interactive music editing can be enabled by incorporating human guidance in the sampling procedure through the conditioning prompt of unmasked tokens. Because our approach isn't conditioned on any signal other than the input audio itself, we find that various types of prompts are useful for obtaining coherent samples, as they lower the amount of multimodality when sampling from the model. Like AudioLM, we can prompt our model with prefix audio of some duration (usually

---

[5]$k = \gamma(\frac{t}{t_T})D$, where t is the current iteration, t_T is the total number of iterations, and D the total number of tokens in the sequence. The scheduling function γ is a cosine schedule.

between 1 and 4 seconds), and it will provide a continuation of that audio. Unlike AudioLM, and other auto-regressive approaches, we can also prompt our model with suffix audio, and it will generate audio that leads up into that suffix. We can provide prefix and suffix audio, and the model will generate the remaining audio, such that it is appropriate, giventhe specified prefix and suffix.

We can also apply a "periodic" prompt, where all but every Pth timestep are masked.The lower P is, the more the generated audio will sound like the original, as the model is highly conditioned. For example if $P = 2$, then the model is essentially behaving like a upsampler, imputing the tokens for every other timestep. As P increases, the model shifts from behaving in a *compression* mode to a *generative* mode, creating variations that match the style of the original.

Another useful style of prompt are "compression" prompts, where all codebooks other than the most coarse-grained are masked. This gives the model strong conditioning on every timestep, so the model is likely to produce audio that closely matches the original. We can combine this prompt with a periodic prompt with low P for even more extreme compression ratios. Given the bitrate of the codec B, which has number of codebooks N, a downsampling rate P for the periodic prompt, and a number of kept codebooks N_k, we can achieve a bitrate of $B/P(N - N_k)$.

Finally, we can design music-specific prompts, which exploit knowledge about the structure of the music. More concretely, we explore beat-driven prompting, where timesteps that fall on or around the beat are left unmasked. The model is left to create music between these beats, resulting in interesting variations on the original music. These prompts can all be combined to create a very useful music creation tool. In concert with a well designed user interface, VampNet shows promise as the basis for a next-generation music editing and creation suite.

2.3 Experiments

Our experiments aim to evaluate VampNet's capability to both compress and generate music, given the various prompting strategies described in Section 2.2.4. For our objective audio quality measures, we use a multiscale mel reconstruction error and the Fréchet Audio Distance (FAD). Mel-reconstruction error is defined as the $L1$ distance between log-mel spectrograms at various time-

scales,

$$\mathbf{D}_{F,M} = ||\hat{S}_{F,M} - S_{F,M}||_1 \tag{2.3}$$

where F is the FFT size of each spectrogram, and M is the number of mel-frequency bins. We use $F \in [2048, 512]$ and $M \in [150, 80]$, with a hop size of $\frac{1}{4}$ the FFT size. Mel-reconstruction is valuable as a metric for compression quality, but not for generation quality, since it is likely that models produce audio that does not match one to one with the original target audio. For generation quality, we use FAD, which measures the overlap between distributions of real and generated audio. Unlike mel-reconstruction, FAD is geared more towards evaluating if sample quality falls within the data distribution of the real audio, and can be used to evaluate generation quality.

2.3.1 Dataset

Similar to JukeBox [114], we collect a large dataset of popular music recordings. Our dataset consists of 797k tracks, with a sampling rate of 32 khz. These tracks are resampled to 44.1kHz to make compatible with our tokenizer.

2.3.2 Data Preprocessing

We use a subset of 2k tracks for validation, and another subset of 2k tracks for testing. We ensure that there is no artist overlap between train, validation, and test tracks. In addition, we collect a set of music and non-music data (speech, environmental sound), which we used to train our tokenizer, using the datasets described in DAC [58]. All audio is normalized to -24dbFS. We do not use any metadata about these files during training, as our model is trained unconditionally.

2.3.3 Network Architecture and Hyperparameters

The audio tokenizer model we use takes as input 44.1kHz audio, and compresses it to a bitrate of 8kbps using 14 codebooks, with a downsampling rate of 768x. The latent space therefore is at 57Hz, with 14 tokens to predict at every timestep. We designate 4 of these tokens as the coarse
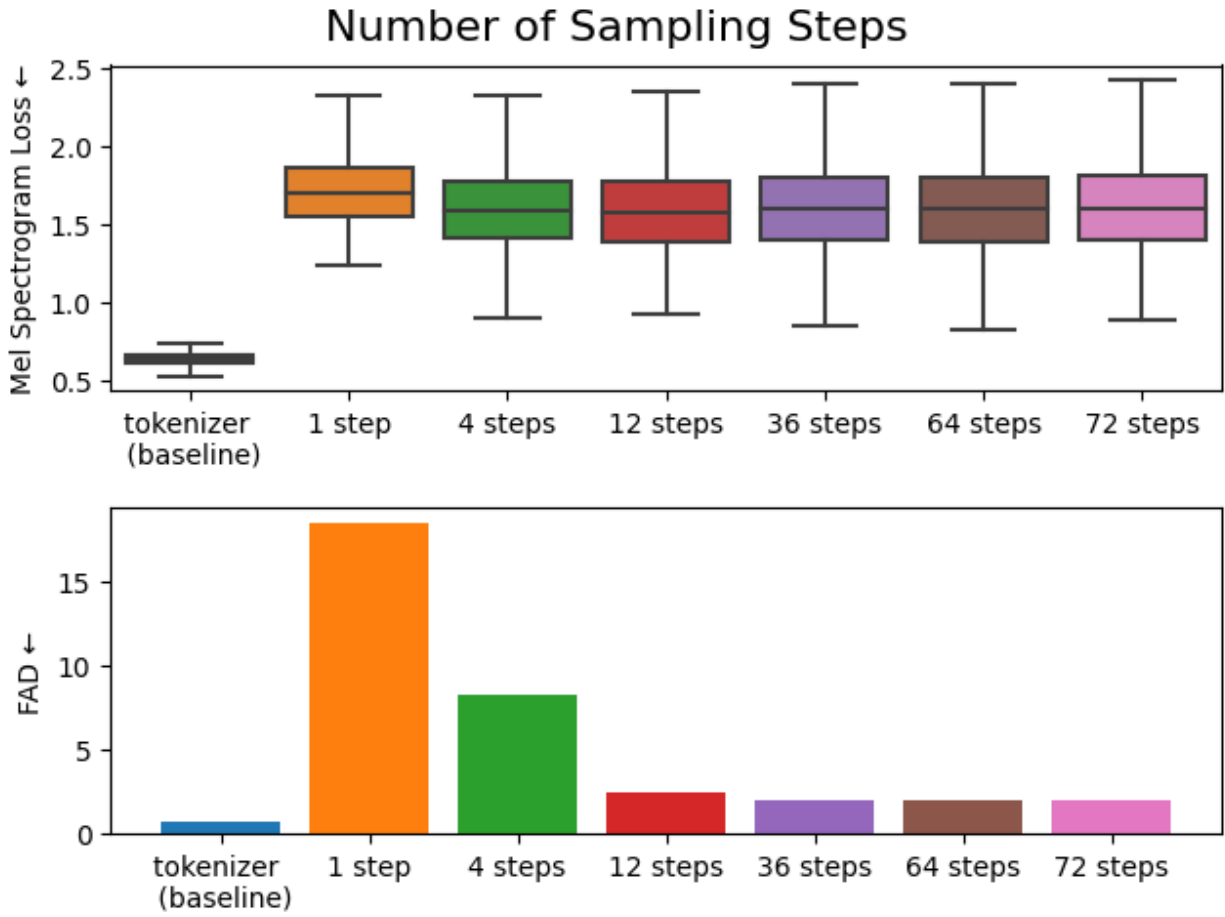
Figure 2.3: Mel reconstruction error (top) and Fréchet Audio Distance (FAD, bottom) for VampNet samples taken with varying numbers of sampling steps, taken using a periodic prompt of $P = 16$. The samples were generated by de-compressing tokens at an extremely low bitrate (50 bps), effectively generating variations of the input signals.

tokens, and the remaining 10 as the fine tokens. Refer to the Descript Audio Codec [58] for details on the tokenizer architecture. We train the tokenizer for 250k steps.

The VampNet architecture (for both coarse and coarse-to-fine models) consists of a bidirectional transformer [115] with relative attention [125] and an embedding dimension of 1280 and 20 attention heads. The coarse model has 20 attention layers, while the coarse-to-fine model has 16. We train the coarse and coarse-to-fine model for 1M and 500k steps, respectively. We train with the AdamW optimizer [126] with β_1 and β_2 set to 0.9 and 0.999, respectively. We use the learning rate scheduler introduced by Vaswani et al [115] with a target learning rate of 0.001 and 10k warmup steps. We use a dropout of 0.1, and a batch size of 25, with a GPU memory budget of 72GB.

2.3.4   Efficiency of VampNet

We first validate that VampNet can generate realistic music audio in a low number of steps. To do this, we run VampNet using one of our prompts (the periodic prompt, with $P = 16$) on our test set, on 10-second excerpts. We vary the number of sampling steps in $[1, 4, 8, 12, 36, 64, 72]$, and report metrics for each sampling step.

2.3.5   Effect of prompts

We seek to understand how VampNet responds to different prompts, as discussed in Section 2.2.4. The prompts range from "compression" prompts, which compress music to a low bitrate, to more creative "generative" prompts. We examine whether compression and generative prompts exist on a continuum, and whether decompression from low bitrates results in generative behavior.

We draw 2000 10-second examples from our evaluation dataset, encode them into token streams with our audio tokenizer, and manipulate the token streams in four ways:

1. Compression prompt: C codebooks are left unmasked, starting from the coarsest codebook. All other tokens are masked. We set $N_k = 1$.

2. Periodic prompt: every Pth timestep is left unmasked. In an unmasked timestep, tokens

41
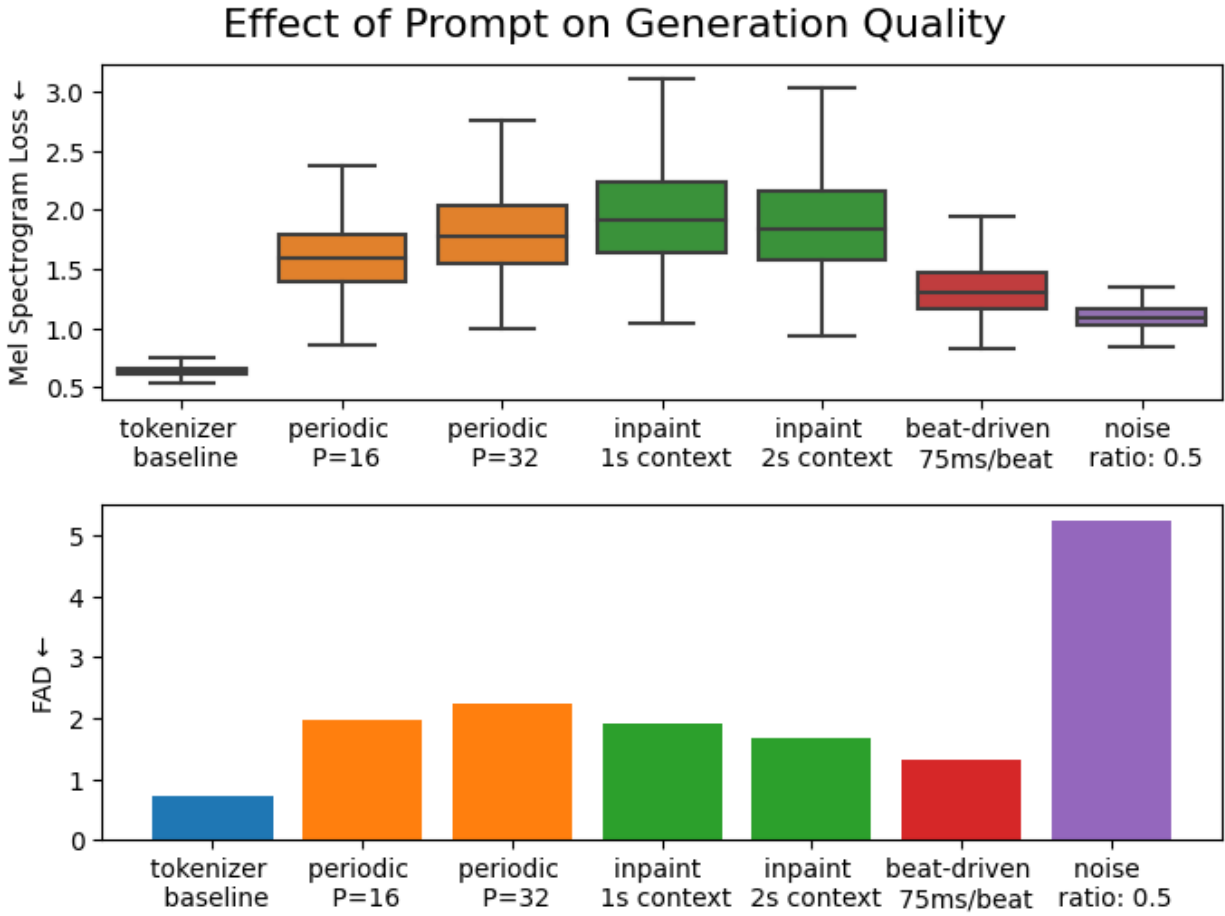
Figure 2.4: Multiscale Mel-spectrogram error (top) and Fréchet Audio Distance (FAD, bottom) for VampNet 10s samples taken with a different types of prompts.

from every codebook are unmasked. All other tokens (e.g. tokens in timesteps that do not correspond to the period P) are masked. We set $P \in [8, 16, 32]$.

3. Prefix and suffix (inpaint) prompts: a segment at the beginning and at the end of the sequence is left unmasked. All other tokens are masked. This prompt is parameterized by a context length in seconds. We set the context to be either 1 second or 2 seconds, which corresponds to 57 or 114 timesteps.

4. Beat-driven prompt: we first process the audio waveform with a beat tracker [127]. Then, around each detected beat, we unmask timesteps to the right of the beat. We examine a 75ms unmasked section around each beat, which is about 4 timesteps per beat.

After manipulating the input token streams with our prompts, we generate new musical signals from these masked token streams using VampNet, and compute FAD and mel-reconstruction error between the generated signals and the input signals from our music dataset. We include a noisy token stream baseline, where a portion (as dictated by mask ratio r) of the tokens in the input token stream are replaced with random tokens. We also include as baseline the codec by itself, as well as the coarse-to-fine model.

Finally, we examine how these prompts can be combined - specifically the compression and periodic prompts. By manipulating the hyperparameters of these prompts (C and P), we can shift the model behavior from compression to generation. As more timesteps are masked, the model must generate plausible musical excerpts that connect the unmasked timesteps, that may not match the input music.

2.4 Results and discussion

Results for our experiment varying the number of sampling steps used to generate samples with VampNet are shown on Figure 2.3. We find that VampNet achieves the lowest FAD with 36 sampling steps, although 12 sampling steps achieves comparable performance. In practice, we find that samples taken with 24 steps achieve a fair trade-off between generation quality and compute speed, with 10-second samples taking around 6 seconds to sample on an NVIDIA RTX3090. In

contrast, to generate 10 seconds of audio with an autoregressive model would require 574 steps, which would take around 1 min to generate 10 seconds of audio, given an autoregressive model with the same number of parameters as ours, and the same tokenizer.

Results for our study on the effect of each prompt are shown in Figure 2.4. First, we note that while the noisy token baseline has comparable mel reconstruction to all prompts, it performs very poorly in terms of FAD. This indicates that while our prompting strategies may result in audio that is not a perfect match to the original input audio, it still falls inside the distribution of plausible music.

Of our proposed prompts, we find that beat-driven prompts perform best, achieving the lowest FAD of all prompts. A notable result here is that the periodic prompt with $P = 16$ (35 conditioning timesteps) performs on par with inpainting with 1 second of context (57 conditioning timesteps). Therefore, prompt techniques that spread out the conditioning tokens throughout the sequence (periodic prompts) are able to use fewer conditioning timesteps to generate samples of comparable quality to those generated by sampling techniques that place all of the conditioning tokens at the start and end of the sequences (inpainting).

Qualitatively, we also find that beat-driven prompts can keep a steadier tempo than other prompts, though their outputs tend to resemble the original music closer than periodic prompts. In practice, a mix of beat-driven, periodic, and inpainting prompts can be employed to steer of VampNet in creative ways. To illustrate, we highly encourage the reader to listen to the accompanying sound samples [6].

We then combined periodic and compression prompting to show how the model's behavior shifts between reconstruction and generation tasks, as more tokens are masked away. Results for this experiment are shown in Figure 2.5. At higher bitrates, (600 bps and above), VampNet is able to accurately reconstruct the original music signal, achieving low mel-spectrogram error and FAD values with respect to the evaluation music audio. At bitrates of 200bps and below, VampNet has comparable reconstruction quality to the noisy token baselines, indicating that the sampled

---

[6]audio samples: `https://tinyurl.com/bdfj7rdx`
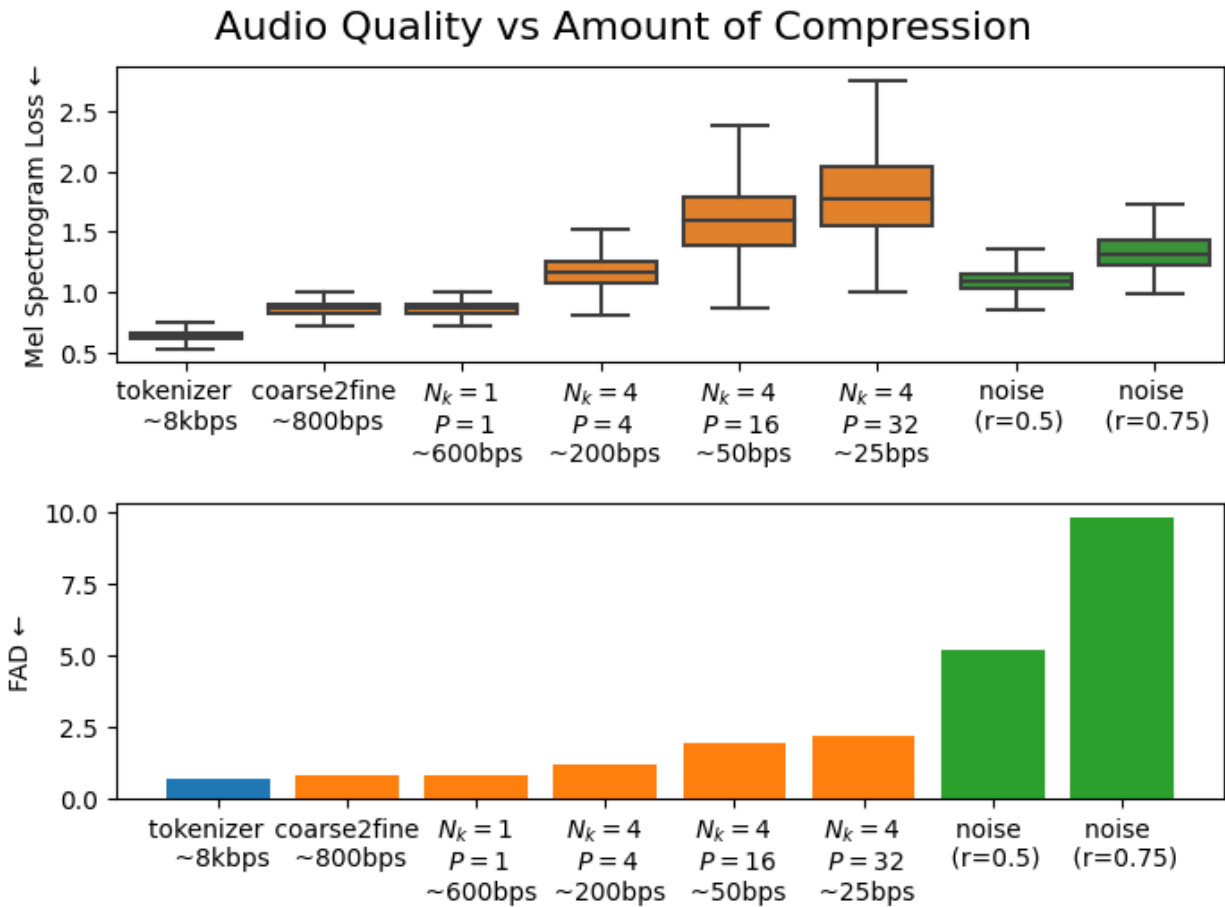
Figure 2.5: Mel-spectrogram error (top) and Fréchet Audio Distance (FAD) (bottom) for VampNet samples at varying bitrates. A baseline is provided by replacing tokens in the input sequence with random tokens, per noise ratio r.

VampNet signals no longer resemble the input audio in terms of fine-grained spectral structure. However, the FAD for VampNet samples at low bitrates is much lower than the FAD for noisy baselines. This indicates that even though VampNet isn't able to reconstruct the input music signal at low bitrates, it is still able to generate coherent audio signals with musical structure, that are closer to the distribution of "real music" than our noisy baseline.

2.4.1 Ethical Considerations Surrounding VampNet

The dataset used to train VampNet was collected by me during the months of September and October 2023, for research purposes at the Interactive Audio Lab. The dataset contains music from approximately 206k artists across approximately 5.8k genres. A list of genres was copied from the Echo Nest's Every Noise at Once [7] genre map. This list of genres was used to collect Spotify API metadata (no audio) for 797k tracks. The audio for each track was then downloaded using the open source tool spotdl [8], which uses the collected Spotify track IDs to find a corresponding YouTube video and download the track's corresponding mp3 recording.

Admittedly, the data used to train the VampNet model contains copyrighted recordings. It's worth noting that this data was acquired (and is used) for the purpose of academic research only. Scraping copyrighted data to train generative models can be extremely harmful and have negative ethical consequences when said data (or resulting models) are used for commercial purposes. Members of my lab (including myself) have taken several measures (both systematically and interactionally) to ensure that the uses of the data collected pose more benefits than harmful risks for the musicians and sound artists at stake.

Barnett [128] identifies 5 broad negative impacts of AI models in music: (1) loss of agency and authorship (2) creativity stifling (3) predominance of western bias (4) copyright infringement and (5) cultural appropriation.

As an active, working musician in Chicago, I've had the opportunity to have casual conversations with my musician friends. I've thought deeply about the impacts highlighted by Barnett, and

---

[7] https://everynoise.com/engenremap.html
[8] https://github.com/spotDL/spotify-downloader

have made several design decisions that mitigate these issues.

Regarding a loss of agency and authorship (1), Barnett refers to Frid [129], who finds that musicians were wary of giving a machine too much control. Using the techniques discussed in Chapter 5, VampNet allows for gestural interactions with fine-grained temporal detail, giving the musician a greater sense of control over the temporal unfolding of a sound than, say, a text-to-audio model.

I believe the issue of generative models stifling creativity (2) becomes a problem when the generative model itself is encapsulated in an interface made for casual, quick creation and consumption of music, where the user is put into a curator mindset rather than a creator one. This interaction is often commodified into a pay-to-play product, e.g., `suno.ai` and `udio.com`. Instead, I try to position VampNet as a computer music tool for sound transformation and collaging instead (Chapter 5), meant to be used as part of a larger musical process, like a live performance with a lead instrument (Section 5.4.2), an interactive sound installation (Section 5.4.4) or an immersive fixed-media electroacoustic voice transformation piece (Section 5.4.3).

Fine-tuning a generative model's weights on a new collection of sounds (Section 5.1) makes it extremely unlikely that samples from the original model will contain sounds from pretrained model's training data. I highly encourage (and facilitate) that musicians use their own collections of sounds (either recorded or hand-picked by them) to use as material for VampNet using a method called sound palette fine-tuning, described in Section 5.1. A musician bringing their own sound palette to the model mitigates the predominance of western bias (3), copyright infringement (4), and (5) cultural appropriation. By consciously curating their own sound palettes, musicians can choose to collect sounds that do not reflect western biases, infringe copyright or harmfully appropriate any cultural elements.

Using the Data for Good: Training Data Attribution for Generative Models

A generative modeling scenario would **not** be the first time a musician copied another musician's recording to use as material in a new original composition. In fact, entire musical styles and genres

are based around this technique, **sampling**.

A big, great contrasting difference between sampling and generative modeling here is the sound artist's knowledge of the provenance of the material being copied/sampled/generated. Artists who sample recordings, most of the time, are fully aware and knowledgeable of the origin, style, original artist, and musical tradition that a particular sample may belong to.

A generative modeling user, as it currently stands, unfortunately does not have any access to this information when they generate sonic material with a generative model: the generative model is a black box that does not elucidate its influences when generating a particular piece of music.

My labmate, J. Barnett, led a study [130] (in collaboration with me and Bryan Pardo) which proposes a method that leverages VampNet's training dataset (as well as a VampNet model itself) to **inform** the user of a particular generation's "influences": the songs in the training dataset which are the most similar to a piece of generated audio. This method requires us to have full access to a large-scale generative audio model as well as the data used to train it. By attributing each generated snippet to its closest audio in the training data, we can create generative music co-creation systems that encourage a musician to gain knowledge of the surrounding musical context and culture of the music being created, which is an important duty of any musician working within any tradition.

2.5   Impact and Follow On

VampNet set the stage for other research works that explored music generation via masked/parallel acoustic token modeling, including [70, 131, 57, 132]. My technical work with VampNet was followed by a period of creative practice and collaboration with sound artists, composers and performers, interface development, and further technical development, as discussed in Chapter 5.

From a technical standpoint, nowadays, continuous audio latent diffusion models are considered an (*arguably* better) alternative to acoustic token modeling methods, which can be considered a form of discrete diffusion [74]. Continuous audio latent diffusion models are less unwieldy than discrete audio token diffusion models, as they require no residual vector quantization on the latent audio encoder.

That being said, VampNet allows for manipulating audio in particular ways that cannot be replicated in (unless they adopt a training scheme similar to VampNet). I propose and discuss these new token manipulation techniques and how they can be used in creative works in my practice-based research chapter (Chapter 5).

The next chapter (3) discusses Sketch2Sound, a system that is capable of synthesizing audio from vocal (and sonic) imitations and interpretable, time-varying control signals. The work presented in this makes use of an audio latent diffusion model.

CHAPTER 3

CONTROLLABLE AUDIO GENERATION VIA CONTROL SIGNALS AND SONIC IMITATIONS
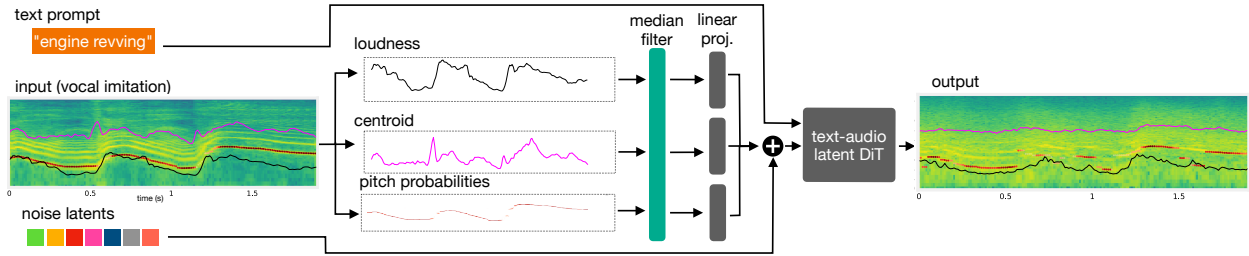
Figure 3.1: Overview of Sketch2Sound. We extract three control signals from any input sonic imitation: loudness, spectral centroid (i.e., brightness) and pitch probabilities. We apply median filters to these signals, encode them via a linear projection, and add them to the noisy latents that are used as input to a DiT text-to-sound generation system. To hear this example (and many more) go to https://hugofloresgarcia.art/sketch2sound.

3.1 Prologue

This work was completed in the Summer, Fall of 2024 and Winter 2025 during a Research Internship at Adobe Research in collaboration with Oriol Nieto, Justin Salamon, Bryan Pardo, and Prem Seetharaman. Thus, uses of the pronoun "we" in the remainder of this chapter refer to work led by me, in collaboration with the researchers mentioned above. This work was previously published at the ICASSP 2025 conference [17].

It should be noted that this system was announced by Adobe at the Adobe MAX conference in October 2024 as part of a suite of tools for sound design for video, under the name "Project Super Sonic" [1]. As of June 5th, 2025 a beta version of a voice-to-sound effects interface has been announced as coming to Adobe Firefly [2], a generative modeling playground for creating images, audio, and videos with generative modeling.

---

[1]`https://techcrunch.com/2024/10/15/adobes-project-super-sonic-uses-ai-to-generate-sound-`
[2]`https://firefly.adobe.com/`

3.1.1 Individual Contributions

I (H. Flores García) proposed the initial motivation to generate sounds from vocal imitations, inspired by discussions with P. Seetharaman and my compositions where I transformed a collection of sounds with vocal gestures using VampNet (See Section 5.4). P. Seetharaman and I devised the original "frame" of how the system would work as a "simplest solution", which would end up working quite satisfactorily. P. Seetharaman, O. Nieto, J. Salamon, and B. Pardo all provided guidance on the research methods and suggested compelling evaluations.

From a technical standpoint, I (H. Flores García) wrote all the code for the Sketch2Sound method, adapting off of the baseline text-to-sound DiT recipe that was created and maintained by P. Seetharaman, O. Nieto, and J. Salamon. This includes the control signal extraction, median filtering, adapter layers, fine-tuning, and all the required evaluations metrics. The paper was primarily written by H. Flores García and edited by all the other authors.

3.1.2 A remark on the motivation behind this work

All the other chapters in this dissertation (except for this one) are focused on systems for making experimental music and sound art.Instead, this chapter is motivated around an adjacent yet slightly different craft: sound design (specifically Foley sound). Unlike sound artists or computer musicians, sound designers often work very closely with other media, like visual arts, video, film, or video games. This functionally changes the role of sound in the resulting artwork, and thus also changes what the most suitable instrument for that craft might look like. However, there's a great deal of similarity between experimental music/sound art and foley sound: **both crafts value gestural expression**. Improvised experimental musics often incorporate the concept of gesture, exploring and playing with the gestural limits of the performer's instrument. Foley sound is a *performance art* that happens behind the scenes: Foley artists play their props *like musical instruments*. Foley artists highly value the "simple, beautiful, and performative nature"[3] of sound design.

---

[3] https://www.youtube.com/watch?v=WFVLWo5B81w

We, as designers and engineers of sound design tools, must follow these design principles and create systems – much like musical instruments – that support and enable new kinds of "simple, beautiful, and performative" interaction.

3.2 Introduction

Sound design is the craft of storytelling through sonic composition. Within sound design, Foley sound is a technique where special sound effects are designed and performed in sync to a film during post-production [133]. These sound scenes are typically performed by a Foley artist on a stage equipped with abundant sound instruments and other soundmaking materials[4]. Foley sound is a skilled and gestural performance art: performing a sound scene with sound-making objects and instruments (instead of arranging pre-recorded samples post hoc) allows sound artists to create fluent and temporally aligned sounds with a "human" (i.e., gestural) touch. Adding this gestural touch to the resulting sound composition often results in a sonic product of great aesthetic and production value.

Recent research in generative modeling for sound has paved the way for text-to-sound systems[65, 63, 53], where a user can create sound samples from text descriptions of a sound (e.g., "explosion"). While the text-to-sound paradigm can help a sound designer find sounds more quickly (and, perhaps in the future, with a higher degree of specificity), a sound designer still has to painstakingly modify the temporal characteristics of the generated sound so that they can be in sync with the visuals in the editing timeline. This is in opposition to the natural way that Foley artists gesturally create sound effects by physically performing with physical soundmaking objects.

To overcome the drawbacks of a purely text-to-audio interaction, several works in the music domain sought to condition generative models on audio [16], parallel instrument stems [70], melody [56], sound event timestamps and frequency [134], or multiple structural control signals like song structure and dynamics [72]. Notably, [92] condition an audio VAE on control signals such as brightness and loudness, though their experiments are limited to models trained on nar-

---

[4]Example of a Foley artist performing a scene: `youtu.be/WFVLWo5B81w`

row sound distributions (e.g., violin, darbouka, speech) and not a multi-distribution text-to-audio model. For speech, [85] proposes a fully interpretable and disentangled representation for speech generation and editing, which allows for fine-grained control over the pitch, loudness, and phonetic pronunciation of speech.

The human voice is a gestural sonic instrument [4]: it allows us to realize sounds without having to perform any symbolic abstraction (i.e., putting a sound into words) beforehand. When humans communicate audio concepts to other people, they typically combine language and vocal imitation [5, 6, 7], and recent work has shown its utility for query-by-example search of audio databases [**zhang2016imisound** , 8, 9]. This is a more natural method than describing the evolution of pitch, timing, and timbre via pure text descriptions [5], and voice-driven sound synthesis interactions have been of interest long before modern generative modelling for their embodied capabilities [135, 136, 137, 138, 139, 140].

We propose Sketch2Sound: a text-to-audio model that can create high-quality sounds from sonic imitation prompts by following interpretable, fine-grained time-varying control signals that can be easily extracted from any audio signal at different levels of temporal detail: loudness, brightness (*spectral centroid*) and pitch. We expand upon previous work [71] by developing a method capable of following the loudness, brightness *and* pitch of a vocal imitation, with the option to drop any of the three controls. Additionally, we propose a technique that varies the temporal detail of the control signals used during training by applying median filters of different window sizes to the control signals before using them as input. This allows sound artists to specify the degree of temporal precision to which a generative model should follow the specified control signals, which improves sound quality in sounds that may be too hard to perfectly imitate with one's voice.

This method is not limited to just vocal imitation: any kind of sonic imitation can be used to drive our proposed generative model – we place the focus on vocal imitation due to people's innate ability to imitate sounds with our voices. Vocal imitations can always be augmented through other sonic gestures like clapping, tapping, playing instruments, etc. Sketch2Sound can be added to any

existing latent diffusion transformer (DiT) sound generation model with as little as 40k fine-tuning steps. Unlike ControlNet methods [73, 141] that require an extra trainable copy of the entire neural network encoder, Sketch2Sound requires only a single linear layer per control.

Our experiments show that Sketch2Sound can generate sounds that closely follow the input control signals (loudness, spectral centroid, and pitch/periodicity) from a vocal imitation while still achieving a high degree of adherence to a text prompt and an audio quality comparable to the text-only pre-trained model. We show that our median filtering technique leads to improved audio quality and text adherence when generating sounds from vocal imitations. We also show that, during inference, a user can arbitrarily specify a degree of temporal detail by choosing a median filter size, allowing them to navigate the trade-off between strict adherence to the vocal imitations and audio quality + text adherence.

To the best of our knowledge, this is the first sound generation model capable of following vocal imitations *and* text prompts by conditioning on a set of holistic control signals suitable for generating sound objects with fine-grained, gestural control of pitch, loudness, and brightness. We believe Sketch2Sound will give sound artists a more expressive, controllable, and gestural interaction for generating sound-objects than existing text-to-audio and other conditional sound generation systems. **We highly encourage the reader to listen to our audio examples demonstrating Sketch2Sound.** [5]

3.3 Method

We propose a method for conditioning an audio latent diffusion model on a set of interpretable, time-varying control signals that are suitable tasks creating variations of sounds and generating new sounds expressively via text-prompted sonic imitations.

3.3.1 Time-varying control signals for sound objects

We use the following control signals as conditioning for Sketch2Sound:

---

[5]`https://hugofloresgarcia.art/sketch2sound`

- **Loudness:** We extract the per-frame loudness of an audio signal by performing an A-weighted sum across the frequency bins in a magnitude spectrogram [85] and taking the RMS of the result.

- **Pitch and Periodicity:** We use the raw pitch probabilities of the CREPE [142, 143] ("tiny" variant) pitch estimation model. To avoid leaking timbral information in this signal, we zero out all probabilities below 0.1 in the pitch probability matrix.

- **Spectral Centroid** is defined as the center of mass of the frequency spectrum for a given audio frame. Frames with a higher centroid will be perceived as having a brighter timbre. To preprocess the centroid, we convert the signal from linear frequency space (i.e., Hz) to a continuous MIDI-like representation, scaled to roughly a $(0, 1)$ range by dividing the input signal by 127 (note G9, roughly 12.5kHz), which we found to stabilize the first steps of training.

Other momentary time-varying control signals may be used as well.

3.3.2 Conditioning a latent audio DiT on time-varying control signals

Refer to Figure 1 for a visual overview of our approach. We use a large pre-trained text-to-sound latent diffusion transformer (DiT), similar to the one described in [65, 62] (text-conditioned only, no timing conditioning) and adapt it to generate sounds conditioned on the time-varying control signals mentioned above. The latent diffusion model for text-to-sound generation has two parts: first, a variational autoencoder (VAE) compresses 48kHz mono audio to a sequence of continuous vectors of dim 64 at a rate of 40Hz. Then, a transformer model is trained to generate new sequences of latents, which can be decoded into audio using the VAE decoder. This text-to-audio DiT was pre-trained on a large mix of proprietary, licensed sound effect datasets and publicly available CC-licensed general audio datasets. Once the model is pre-trained, we fine-tune it for 40k steps and adapt it to handle our time-varying control signals.

Because the time-varying control signals can be easily and efficiently extracted from any audio signal on the fly, we can fine-tune the pre-trained text-to-audio model in a self-supervised manner:

Given any input audio signal, we extract the three control signals (loudness, centroid, pitch) from the audio signal and use them as conditioning for the model during fine-tuning. The model is then fine-tuned using the same recipe used during training: learning the reverse diffusion process from a set of noisy latents with text conditioning, along with our proposed control conditioning.

To align the time-varying control signals with the latents from our text-to-sound DiT, the control signals must be extracted at the same frame rate as the audio VAE latents or interpolated to this frame rate. This allows us to perform a simple conditioning method: condition a latent diffusion model ϵ_θ by simply adding a linear projection layer from our control signals to the noisy latents used as input to the diffusion model. Since these time-varying control signals are highly localized to their given time frame, a simple linear layer suffices to incorporate each time-varying signal as conditioning to the model.

Given the noisy latent vector sequence $\mathbf{z} \in \mathbb{R}^{D \times N}$ that is used as input to a latent diffusion model ϵ_θ with embedding dimension D and sequence length N, we introduce our time-varying conditioning signal $\mathbf{c}_{ctrl} \in \mathbb{R}^{K \times N}$ with dimension K and sequence length N to the latent $\mathbf{z}$ by applying a trainable linear projection layer $p_\theta(\mathbf{c}_{ctrl}) \in \mathbb{R}^{D \times N}$ to the input conditioning, and adding the result directly to the latents used as input to the diffusion model: $\mathbf{z}_{ctrl} = p_\theta(\mathbf{c_{ctrl}}) + \mathbf{z}$. We can repeat this process for any number of time-varying control signals that we'd like to condition our latent diffusion model.

During fine-tuning, the loss configuration for the model does not change from the one used in original training; that is, we do not apply any reconstruction losses for the control curves themselves, removing the need to pass through the VAE decoder during fine-tuning. Despite not measuring loss on the control signals, providing them as input during fine-tuning is sufficient for the model to condition generation on them. As a result, these signals become useful for control at inference time. To ensure we can generate without requiring all the control signals, we perform dropout during fine tuning on the control signals by zero-ing out the control embeddings $p_\theta(\mathbf{c_{ctrl}})$ before they are added to the diffusion model's latents $\mathbf{z}$. We drop each control signal (as well as the text conditioning) individually with a 20% probability, with an added 20% probability of dropping

out all signals together. Likewise, these signals can be dropped out at inference if a user wishes to control only one or two controls while omitting others.

At inference time, we follow the two-conditioning classifier-free guidance setup described in [144]. We use guidance scales s_{ctrl} and s_{text}, which can be used to trade off the guidance strength for the control signal conditioning and text conditioning, respectively. We find that using a single guidance scale for all three control signals together (s_{ctrl}) is a sufficient approach, but future work may explore the effect of applying guidance strengths to each time-varying control independently of each other. Anecdotally, we find that setting s_{text} to a value of 5 and s_{ctrl} to 1 achieves results with good text adherence while following the contour provided in the sonic imitation controls.

3.3.3 Creating sketchlike controls via control-rate filtering

Even though the human voice is remarkable at imitating sounds, there will still be a mismatch between the control signals of the target sound and the control signals of the vocal imitation. To overcome this issue, we propose a technique to make the controls sketchlike by applying random median filters to the control signals at different window sizes (1-25 control frames) before they are used as input to the model. This filtering technique can help mitigate the mismatch in temporal specificity between the vocal imitation and target sound and help the model produce higher-quality sounds from sketchlike vocal imitations. Our experiments show that during inference, a sound artist is free to adjust the control rate of their input control signals, giving them an interpretable control over the trade-off between text-prompt adherence and fine temporal precision.

3.4 Experimental Design

Our experiments evaluate Sketch2Sound's ability to synthesize high-quality sounds from vocal imitations. Except for the text-only baseline (which doesn't need fine-tuning), we fine-tune every model for 40k steps with the same configuration used for training. For all fine-tunings, we use the text-only baseline as the starting checkpoint.

The main dataset used for evaluation is VimSketch [145], which consists of approximately

57

12k vocal imitations, each with a text description and reference sound. For each model variant, we generate 10k examples with durations of up to 5 seconds using the vocal imitation and text description as conditioning. We evaluate Sketch2Sound along the following characteristics:

Audio Quality: To measure a model's ability to synthesize high-quality sound effects, we compute the Frechét Audio Distance (FAD) [146] using a proprietary dataset of 40k high-quality sound effects as the reference set, and 10k sounds generated from vocal imitations from the VimSketch dataset as the evaluation set, as suggested by [147]. We report the FAD for VGGish [148] and LAION-CLAP [149] embeddings. **Text Adherence:** We measure how well our generated audio adheres to the target text prompt by computing the CLAP embedding cosine similarity [149] between audio generated from a sonic imitation and the target text prompt for every example.

Control Signal Adherence: Finally, to measure the adherence of the generated audio to the vocal imitations, we measure the error (L1) between the input and generated control signals (loudness, centroid, pitch) only on non-silent (loudness $> -40dB$) frames. We measure loudness error in dBFS RMS and centroid error in semitones (st). We report the following pitch metrics: pitch error (st), chroma error (from the predicted pitch), and periodicity error, as estimated by torchcrepe [143]. We only measure pitch and chroma error on *voiced* frames, i.e., where the periodicity predicted by torchcrepe is greater than a threshold of 0.5 for both the vocal imitation and the model output.

3.5 Experiments

3.5.1 Control signals

First, we validate that Sketch2Sound can synthesize sounds from a reference vocal imitation and a text prompt while achieving a competitive audio quality, text adherence, and adherence to the vocal imitation. We fine-tune three models, each with a different set of control signals (loudness only, loudness+centroid, loudness+centroid+pitch), and compare the performance of these models, along with a text-only baseline, using the metrics discussed in Section 3.4. For all model variants, we use a fixed median filter window size of 10 at inference.

3.5.2 Sketch type ablation

To observe the effect of our random median filtering as a way of creating sketchlike controls, we compare our approach to a no-filter baseline, as well as an alternative approach using low-pass filters instead of median filters to remove fine temporal detail from the controls during training. We train our model with random median filters with window sizes ranging between 1 and 25 control frames (1 frame = 25 ms). At inference, we use a fixed median filter size of 10. For our low-pass filter approach, we apply random low-pass filters ranging from 5Hz-20Hz and use a fixed cutoff (10Hz) at inference. Likewise, we compare each variant in this experiment in terms of audio quality, text and control adherence, following Section 3.4.

3.5.3 Inference-time control rates

To verify whether our median filtering approach allows for an inference-time trade-off between text adherence and fine-temporal control, we observe the performance of our model trained with random median filters at different inference-time temporal resolutions. Specifically, we generate samples using our model at different inference-time median filter sizes of $\{1, 5, 10, 15, 20, 25\}$ observe their performance on the metrics (Section 3.4).

Table 3.1: Control Signal Evaluation and Sketch Type Ablation (Control Adherence).

	Control Signal	Sketch Type	Control Adherence (Error) $\downarrow$			
			RMS (dB)	Centroid (st)	Pitch (st)	Chroma (st)
control signals	text-only	median (sz 10)	13.41	10.34	13.91	2.96
	loudness (ldns)	median (sz 10)	3.88	10.37	12.45	2.96
	ldns+centroid	median (sz 10)	3.60	4.39	11.17	2.87
	ldns+centroid+pitch (ours)	median (sz 10)	3.60	4.43	1.49	0.48
sketch types	ldns+centroid+pitch	low pass	2.19	3.33	0.44	0.23
	ldns+centroid+pitch	no filters	1.87	3.21	0.45	0.21

Table 3.2: Control Signal Evaluation and Sketch Type Ablation (Text Adherence and Audio Quality).

	Control Signal	Sketch Type	Text Adherence (CLAP ↑)	Audio Quality (FAD) ↓	
				VGGish	CLAP
control signals	text-only	median (sz 10)	0.273	2.57	0.270
	loudness (ldns)	median (sz 10)	0.230	2.69	0.296
	ldns+centroid	median (sz 10)	0.219	2.67	0.306
	ldns+centroid+pitch (ours)	median (sz 10)	0.211	2.51	0.312
sketch types	ldns+centroid+pitch	low pass	0.166	3.30	0.363
	ldns+centroid+pitch	no filters	0.152	3.53	0.379

3.6   Results and Discussion

3.6.1   Control signals

Tables 3.1 and 3.2 validate Sketch2Sound's ability to synthesize sounds using control signals extracted from a vocal imitation + a text prompt while achieving comparable audio quality to a text-only baseline. We incrementally add each control (loudness, centroid, pitch), and observe each model's performance in terms of text adherence (CLAP score), control signal adherence, and audio quality (FAD).

Conditioning a model on a time-varying control signal (e.g., loudness, centroid or pitch) significantly improves the adherence to that control signal compared to when the model is not conditioned on that control. Since loudness, centroid, and pitch are often correlated in natural sounds, incorporating a single conditioning (i.e., loudness) also slightly improves the control adherence for other control signals.

In terms of text adherence and audio quality, introducing control signals produces a slight decrease in audio quality and text adherence. Empirically, we find that this difference is near negligible when compared to the text-only baseline in most cases. We also find that the quality of the generated audio can be a function of how well the user imitates the characteristics of the target
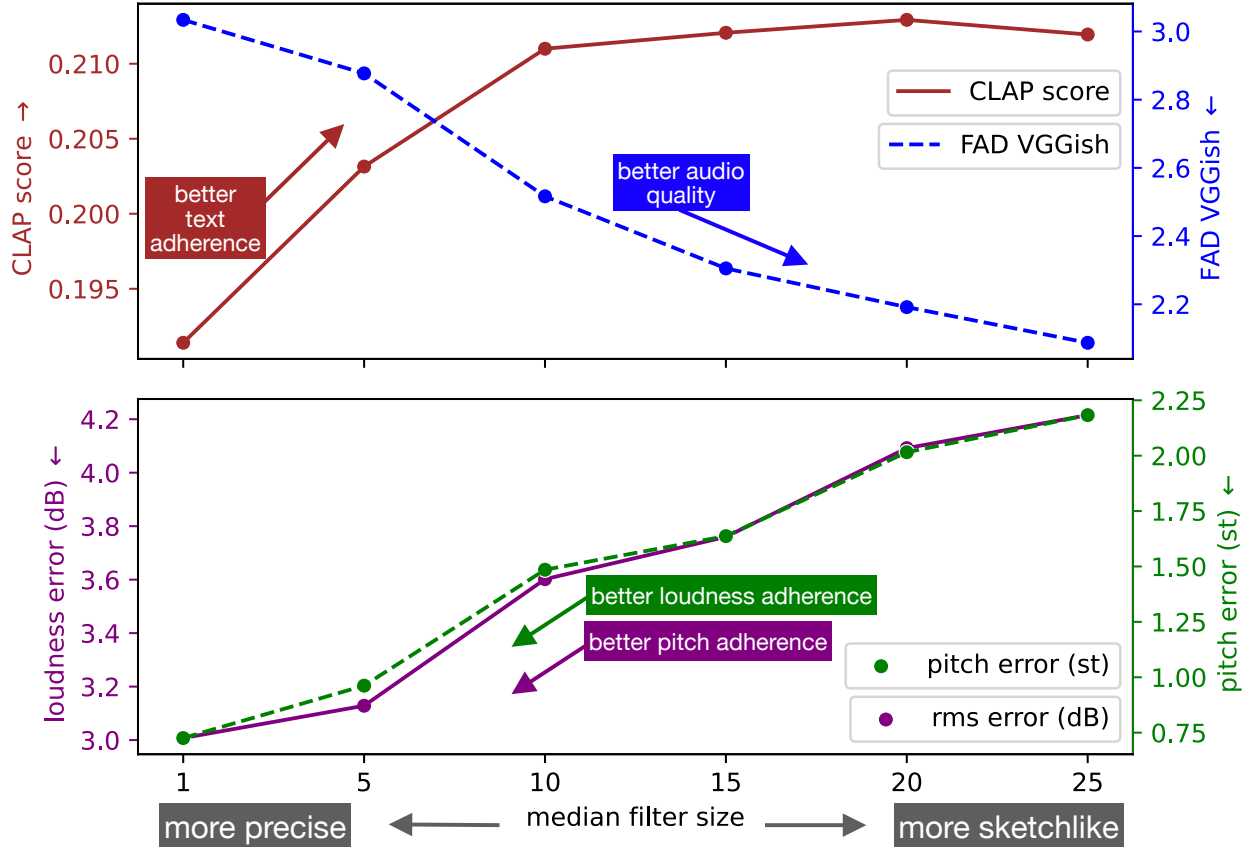
Figure 3.2: At inference, larger median filters are more sketchlike and can lead to higher audio quality, while smaller filters are more precise and may lead to lower audio quality if the vocal imitations aren't precise enough, giving the sound artist a choice over this trade-off.

sound: better-performed vocal imitations lead to higher-quality generated sound effects.

3.6.2 Sketch type ablation

Tables 3.1 and 3.2 show that our median filter method makes Sketch2Sound robust to generating high-quality sound effects from sketchlike vocal imitation control signals, improving the audio quality and text adherence over a no-filter baseline and a low-pass method. Notably, our median filter method is able to achieve a higher CLAP score (text adherence) and lower FAD (audio quality), while trading off the control adherence to vocal imitations. **This trade-off, where lowering the control adherence to vocal imitations improves the audio quality is more desirable than a strict adherence to the controls since most vocal imitations cannot perfectly mimic the fine temporal behavior of target sounds. Generating sounds that exactly follow the vocal**

imitation controls (i.e. *"no filtering"*) results in audio that does not sound like the text, but "speechlike".

3.6.3   Inference-time control rates

Our control-rate filtering method lets a sound artist use different-size median filters at inference time, allowing users to choose the desired amount of temporal detail needed for a particular voice-to-sound example. The results in Figure 2 show that Sketch2Sound can be used with different control-rate resolutions at inference time by using median filters of different sizes. Smaller filters achieve higher control adherence, at the cost of a lower audio quality and text adherence. We hypothesize that the decrease in text adherence and audio quality is due to the mismatch between vocal control signals and the target control signals suitable for generating Foley sounds. However, this flexibility allows one to use smaller filters (i.e., higher temporal resolution) when the vocal imitations are well-performed, and larger filters (i.e., lower temporal resolution) when the vocal imitations are impossible to precisely imitatate with the human voice.

3.6.4   The semantics of control curves are implicitly modeled
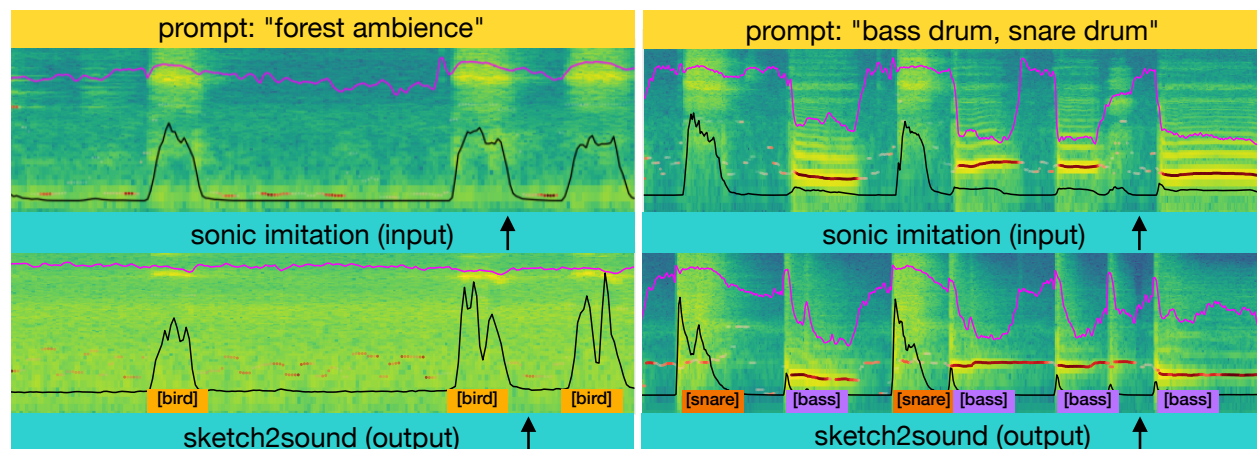
Figure 3.3: (left) When prompted with "forest ambience", bursts of loudness in the controls become of birds without prompting the model to do so. (right) With "bass drum, snare drum", the model places snares in unpitched areas and bass drums in pitched areas.

We find that the control signals can manipulate the semantics of the generated signals. For example, using the text prompt "forest ambience" with a sonic imitation containing random bursts

of loudness in it, we can synthesize bird sounds into those loudness bursts (see Figure 3), without having to use the prompt "birds" at all. The model follows the correlations between bursts of loudness and the presence of birds in recordings of forest ambience in the training data, and so generates bird sounds when prompted with loudness bursts in a "forest ambience" prompt. Likewise, with the prompt "snare drum, bass drum, drum beat", performing a sonic imitation with a series of pitched (bass drum) and unpitched (snare drum) sounds will successfully apply bass drum sounds on pitched regions, and snare drum on unpitched regions. Both of these examples are available on our accompanying website.

3.6.5 Limitations

We found that the centroid control tends to entangle the room tone of the input sonic imitation onto the generated audio. We believe this is because the room tone of the input audio is encoded by the centroid when no sound events are occurring in the input audio. A potential solution is to drop the controls when the signal is quiet, though this may fail with overcompressed inputs or loud room tones.

CHAPTER 4

TWO-STAGE AUDIO GENERATION SYSTEMS AND MUSICAL PRACTICE

4.1 Introduction

Chapters 2 and 3 discussed the technical design of two expressive and controllable generative modeling systems for creating sounds via sonic gestures. However, little has been said (in this dissertation and in the broader literature on large-scale generative music modelling) about how two-stage audio generation systems fit into an existing musical tradition or performance practice. Nor has there been sustained exploration on how these systems could facilitate new and compelling creative interactions beyond casual, consumer-facing interfaces like Suno[1]'s text-to-music (and more recently, MusicFX DJ[2]).

This chapter discusses musical instruments and their inextricable entanglement to their surrounding musical practice. I will also discuss the role of the instrument maker as "the first player" of a new musical instrument, and how engaging in a mixed creative and research practice is a powerful way to evaluate technical advancements and seek new technical and creative directions. I will briefly overview of the aesthetics of Experimental AI music, and discuss the interactional capabilities of generative AI musical instruments and co-creation systems in the context of Curtis Roads' time scales of music [150]. This provides the necessary background material required for Chapter 5, the *neural tape loop*: a new generative musical meta-instrument.

4.2 Why situate within a musical practice

Two-stage generative models, such as acoustic token prediction models (Chapter 2) or acoustic latent diffusion models (Chapter 3), require unprecedentedly large amounts of training data and

---

[1] https://suno.com
[2] https://labs.google/fx/tools/music-fx-dj

compute resources. Thus, these systems are mostly developed by corporations[3] looking to build products for "everyone" – that is, casual consumers, not musicians or artists.

The generated outputs of a model might check all the idiomatic and stylistic boxes from an auditory, perceptual, harmonic, rhythmic and structural perspective – but they lack the social, embodied, and historical context that defines a genre. Musical genre is not just about sonic resemblance or being able to create new chords, melodies, and rhythms within a particular idiom [**todo**] – it's about shared practices, gestures, lineages, participation, and discourse within an artistic community [40]. **A generative music co-creation system's musical validity cannot be solely established by the quality of the audio it produces or the ease of interaction, but by how, where, with whom and why it is used.** For example, although a system like MusicFX DJ is technically capable of generating genre-specific outputs (e.g., blues), you wouldn't bring MusicFX DJ to a blues jam session.

Generative models, like other musical instruments, do not exist in a void. They are inseparable from their contexts – musical, cultural, social, aesthetic – and the traditions in which they are deployed [14]. The aesthetics of a musical style are often a reflection of the affordances of the musical instruments used to make that music. At the same time, a sound artist (musician, instrument-maker, or both) shapes and transforms these instruments to accommodate new expressive needs, enabling deeper and more nuanced engagement with a technique, expanding a particular style, or branching off to a new one.

This pattern can be seen across history. Musical instruments built upon the same core sound-producing system, like a set of strings stretched over a resonating body, have evolved into highly distinct instruments (e.g., violin, oud, steel string guitar, nylon string guitar, sitar, zhongruan, cuatro puertorriqueño, cuatro venezolano, requinto, bajo sexto, electric guitar, to name a few). These variations on the core system have emerged over large periods of time, from a mixture of different sonic goals and an ongoing negotiation between the affordances of the core sound-production system, the physical materials available as well as the aesthetic values of the surrounding cultural

---

[3]suno, udio, musicfx dj

65

context.

The aesthetic content of music is tied to the technical affordances of the instruments that produce it: what an instrument makes easy, difficult, or impossible directly shapes the kinds of musical idioms that emerge through it. The ways in which these technical affordances create particular aesthetic outcomes are thus worth inquiring into.

For example, consider the case of the electric guitar and rock music. The electric guitar was first developed as a technical solution to amplifying a guitar to play with a large live band [151]. However, the technical mechanisms used to amplify said guitar had aesthetic implications of their own – the (originally unwanted) distortion produced by early vacuum tube guitar amplifiers was quickly adopted by artists for its aesthetics, eventually leading to the now ubiquitious sound of overdriven guitar, particularly popular in rock music and its deriving styles. Overdrive impacted the way electric guitarists played their instruments: for instance, overdrive makes chords with complex intervals sound noisy or *muddy* due to intermodulation distortion [152]. The phenomenon of intermodulation distortion led to the establishment of power chords (made up of simple root-fifth intervals) as arguably the main pillar of rhythm guitar in rock music. It is thus undeniable that the aesthetic qualities implied by the electric guitar and its underlying amplification system were of great influence on rock music.

In a similar way, the magnetic tape system served as a foundational technology that gave rise to a myriad of diverse musical practices beyond simple storage and recall of sound waves. Pierre Schaeffer, both a composer and engineer, is a pioneer of the tape splicing, looping, stretching, and pitch shifting techniques that came to characterize Musique Concrète, an experimental music developed in the late 1940s and 50s. Schaeffer's experimental work with Concrète in the 1950s explored the tape manipulation techniques that later adapted, matured, and spread to popular music in the 1960s and 1970s and later into the digital age in the 1980s, eventually evolving into an entirely new family of musical instruments that are now ubiquitous through many genres of popular music, called "samplers".

Experimental music can function as a creative testbed: a space for exploring new instrumental

possibilities and aesthetic frameworks.

4.3 Musicmakers and their instruments

Musicians and their musical instruments have complex, entangled relationships that require more nuanced design frameworks than the traditional device-user framing associated with HCI research. Rodger et al. [14] point out that viewing a musical instrument (or potentially, a co-creative generative musicmaking system, if you will) as a "device" and the performer as a "user" carrying out a functional task has many limitations when designing (and evaluating) the musical instrument. The relationship between a human and their musical instrument is not as simple: framing a musical instrument as a device with an intended use (and thus evaluating it based on how well it carries out that intended use) does not account for the many possible *unintended* uses of the instrument. A great example is the use of extended techniques, like the *col legno* technique associated with bowed string instruments [14], which dates back to at least the 17th century [153]. A design framework based on a device-user view of co-creative musicmaking systems could discourage the development of techniques beyond those intended by the original instrument maker, even though these new emergent techniques could lead to the development of new creative practices of great artistic value. Rodger et al. think of a musical instrument and its performer as forming part of an ecology: the musical instrument is a set of affordances that may change and evolve depending on the performer's agential effectivities, skillful behaviors, background, and ongoing cultural and aesthetic context. It is impossible to evaluate the instrument in a way that could generalize to every musician and performer across a range of musical styles. Instead, evaluating the instrument becomes a matter of specificities, where the dimensions of evaluation are dependent on specificities like a particular musician's style, a particular playing context, or any other aspects of the surrounding ecology.

Given that creating a "generalized", "democratized" musical instrument for everyone is an ill-formed goal (as stated in Chapter 1), *who, then, should the instrument-maker design their instrument for?*

I believe that the first player of any new musical instrument should be its maker. An instrument-maker who is aesthetically invested in the instrument they're building is more likely to shape it towards artistically compelling outcomes. The *de-facto* behavior for most engineers and scientific researchers is to *generalize* – that is, to build systems and interfaces that serve the majority of possible cases. While this can be a powerful approach when applied to many practical problems and tasks in audio signal processing (like speech denoising), when applied to musical instrument design, this approach risks losing the specificities that make a musical instrument meaningful for a specific musical community [14]. Cook [154] holds that 'attempting to build a "super instrument" with no specific musical composition to directly drive the project yields interesting research questions, but with no real product or future direction'. In music, what works "for everyone" often works for no one: designing an instrument without a clear artistic context in mind can lead to systems that are technically impressive but musically inert.

As articulated by Jordà [155], *Digital luthierie*, the craft of creating digital musical instruments, is similar to music creation. The process of creating a digital musical instrument unravels as more than an art than a science [154], with "messy unfoldings" and non-linear trajectories throughout the process [156]. George Lewis states: "Musical computer programs, like any texts, are not 'objective' or 'universal', but instead represent the particular ideas of their creators." [157]. The technical affordances of musical instruments significantly shape their aesthetic possibilities by constraining and directing the performer's potential musical actions. Therefore, instrument makers should remain attentive to the aesthetic dimensions of their instruments as they evolve. Engaging in musical composition and performance with the instrument-in-progress is an optimal way for the instrument maker to be keen on the aesthetics of the instrument: an individual's full-fledged creative practice may offer deeper insight into an instrument's expressive capabilities compared to brief superficial evaluations carried out by a generic population of, say, underpaid Amazon Mechanical Turk workers.

In the end, musical performance is the *ultimate* evaluation of a musical instrument design [158]. Thus, researchers in music generation and human-AI co-creation (and their systems) could benefit

from engaging in a music composition and performance, especially when said musical practice contributes to a larger aesthetic context.

4.4 Practice-Based Research in Computer Music and Digital Musical Instruments

As both the maker of a new generative musical instrument and a creative practitioner working within the traditions of experimental music and sound art, I adopt a practice-based research method: one that consists of treating the act of musical creation itself as a mode of inquiry, allowing aesthetic, historical, technical, engineering and performative dimensions to inform and shape each other in an ongoing, entangled process.

Bulley and Shain define practice-based research as "an original investigation undertaken in order to gain new knowledge partly by means of practice and the outcomes of that practice". [15]. In practice-based research, a researcher supports their contributions by demonstrating creative outcomes which may include artefacts (e.g., images, music, designs, models, digital media, performances, exhibitions, etc.) [15] as well as providing "substantial textual contextualization" of these outcomes in the form of published materials [159].

The interdisciplinary nature of nime [4] design has allowed the community to adopt diverse research practices since its origins in early computer music: research works in nime have taken the form of practice-based research, technical reports, theoretical studies, scientific papers or qualitative research [48]. Elblaus et al. [160] argue that nime researchers should "engage more fully with musical practice and how these interfaces stack up through prolonged use in performance".

Outside nime, the human-computer interaction (HCI) community engages in a method similar to practice-based research referred to as autobiographical design [161]. Autobiographical design is considered useful when the researcher is a *genuine* user of the system (i.e., the system is based on the true needs of the researchers). Among the advantages of autobiographical design is that autobiographical design leverages long-term usage to reveal "big effects" (i.e., the core components that

---

[4]Following Gurevich [48], I refer to the international conference on New Interfaces for Musical Expression as "NIME". On the other hand, I refer to the craft of new musical instrument design and research on the subject in general, or the research which predates the conference in the lowercase form "nime".

make or break a system), and requires "real" systems as opposed to more sandboxed simulations present in controlled evaluations [161].

The practitioner-researcher, both a researcher and active practitioner in a professional setting, undertakes a systematic inquiry that is relevant to their practice and has a direct impact on the practice itself [162]. The practitioner-researcher identifies "researchable problems raised in practice, and responds through practice" [163]. In the context of the arts, the practice-based researcher conducts their research through art-making: "they do not wish to suspend their creative work or allow it to become separate from, or sub-ordinate to, the research activity" [164]. Nime practitioner-researchers often posses a wide range of skills spanning music composition and performance, software development, electrical engineering and interaction design. [165]. Luke Dahl writes: "We cannot divorce our design practice from its application in musical performance, for it is through performance that our ideas, embodied as design prototypes, become testable" [166].

Practice-based research may extend beyond the scope of technical reports or scientific papers. Practice-based works engage with more rounded and situated forms of inquiry – sometimes encompassing historical, critical, cultural, political, and aesthetic dimensions. Gurevich [48] points out that the early written works written by pioneering nime artists like Gordon Mumma, Daphne Oram, David Rosenboom and Michel Waisvisz (*proto*-NIMErs) partook in practice-based research: their works were largely concerened with the "critical, theoretical and historical underpinnings of their practice, as well as reflective accounts of their experiments intented to catalyze future creative endeavors". George Lewis, another trailblazer in modern nime design, writes about his instrument/composition *Voyager*: "This work, which is one of my most widely performed compositions, deals with the nature of music and, in particular, the processes by which improvising musicians produce it. These questions can encompass not only technological or music-theoretical interests but philosophical, political, cultural and social concerns as well. [157]"

Contemporary "state-of-the-art" co-creative AI music systems come from research communities primarily rooted in machine learning research and signal processing. These systems thus reflect the values of these scientific research communities, often prioritizing general usability,

scalability, and novelty of output over aesthetic depth or performative nuance. In contrast, the NIME community has fostered interdisciplinary design strategies that embed musical practice and embodied interaction into the research process itself. I argue that co-creative AI research would greatly benefit from adopting practice-based research approaches — especially those centered on iterative performance-based evaluation, practitioner-researcher entanglement, and ethical ecological deployment within musical communities. Such approaches could lead to more compelling, situated instruments that enable not just the generation of perceptually convincing sound waves, but the formation of new musical practices and cultures.

4.5 Experimental AI Music and Sonic Hauntology

I use the term *experimental* AI music to disambiguate the experimental practice rooted in computer music that encompasses generative musical instrument design and composition (*experimental* AI music) from the now-prevalent commercial artefacts created by text-to-music products offered by large corporations as a way to facilitate "accessible" musicmaking (AI music).

Aesthetically, experimental AI music and sound art pieces are characterized by their eerie, liminal, and uncanny qualities. Experimental AI music works often create interplay between perceptually ambiguous sounds and concepts.

Rubinstein [167] argues that [experimental] AI music is an inherent successor to the brief *sonic hauntology* movement of the 2000s, which consisted of artists like The Advisory Circle, William Basinski, The Caretaker and the Ghost Box record label. Sonic hauntologists employed techniques like mixing, collaging,, and manipulated sounds and textures evocative of the mid-20th century, often transforming these into eerie, dreamlike textures and sonic landscapes.

Experimental AI music works exhibit these hauntological qualities, further intensifying them through the opacity and temporal disjunctions inherent to generative models. Rubinstein notes: "Rather than an alienlike sonic object that radically breaks musical tradition, AI music offers an uncanny recapitulation of it instead" [167]. Sonic Hauntology (and experimental AI music) differ from postmodernism in that they complement ontology rather than oppose it [168]: Experimen-

tal AI music repurposes the technologies built for "hyper-commodified stupefaction" imposed by postmodernity and instead provides an "alternative to late capitalist culture's shallow remixes of the sonic past", undermining, rather than reaffirming, its innocuous bricolage of the past [167].

The term *"AI hauntology"* [168] is used to describe an artistic practice where generative models are used to create "music temporal uncertanties and sonic anachronisms as direct emanations of the algorithm's inner workings". Experimental AI artworks address political, technical and aesthetic narratives that comment on "ghosts" of different kinds – critically reframing a technology that is often seen as "the embodiment of technocapitalist accelerationism and greed for power centralization" [168].

4.6 Generative nimes

To date, the most commonly used generative sound model in musical instruments is RAVE, by Caillon et al. [12]. Built by the ACIDS lab at IRCAM, RAVE models have become the most popular choice for experimental musicians and sound artists working with generative models due to its ability to create sound in realtime on consumer hardware. RAVE is also well-integrated into computer music environments like MAX/MSP and PureData using the nn-tilde package [5], as well as in VSTs like NeuTone [6]. I refer the reader to Chapter 1 for more on RAVE.

Moisés Horta Valenzuela (Hexorcismos) built semilla.ai [18], a musical instrument that connects RAVE latent spaces to ancient Mesoamerican divination through "maíz throwing" technique. Shepardson and Magnusson introduced the Living Looper [90], a live looper that records RAVE latent vectors to create "living" versions of the guitar loops with the hope of creating a co-creative instrument with agential behavior. Visi's Sophtar[21] is a tabletop string instrument that incorporates self-playing modes involving feedback and RAVE model processing. Privato et al. built Stacco [19], a musical instrument that leverages magnetic interactions to drive RAVE models. NeuroRack[7] leverages a controllable RAVE model [92] to situate a generative audio synthesizer in

---

[5] https://github.com/acids-ircam/nn_tilde
[6] https://neutone.ai/
[7] https://github.com/acids-ircam/neurorack

72

the context of modular synthesis, allowing for the model's control signals to be manipulated via control voltage (CV) signals.

4.7 AI music co-creation systems and the time scales of music

Music is a time-based art form. Our experience of sonic phenomena is shaped by the temporal scale in which it occurs. We perceive and interact with sounds at different time scales. Thus, different time scales offer different creative possibilities.

Curtis Roads offers a comprehensive continuum of the time scales of music [150], spanning durations from the *infinite* to the *infinitesimal*. Roads describes 9 time scales of music: the ***infinite*** (a theoretical infinite time span), ***supra*** (a scale beyond the duration of an individual composition, spanning the duration of an artistic movement, for example), ***macro*** (the time scale of musical form, normally measuring minutes or hours), ***meso*** (divisions of form, phrase structures of different sizes, measured in minutes or seconds), ***sound object*** (the concept of a 'note', generalized to include complex and mutating sound events, ranging from a fraction of a second to several seconds), ***micro*** (sound particles stretching down to the threshold of auditory perception, measured in milliseconds) , ***sample*** (individual samples in a digital audio system, measured in microseconds), ***subsample*** (events on a time scale too brief to be recorded or perceived, measured in nanoseconds) , and ***infinitesimal*** (the theoretical time span of a delta function).

Of special relevance to this dissertation are the *macro*, *meso*, *sound object*, and *micro* levels. Especially with modern music technology, musical instruments (and other musicmaking interfaces) are able to operate within different time scales of musical interaction. Like Roads, we will be considering both how these time scales affect musical form itself, but how musical instrument design can shape how we interact with musical instruments, generative models, and other musicmaking systems at different time scales.

Figure 4.1 shows a diagram (a) generative music co-creation systems and (b) musical instruments, organized into their respective time scale of interaction. This work borrows Curtis Roads' classification of time scales of musical structure [150] and reframes it to consider the time scales of

73

... supra macro meso sound object micro sample ...

generative co-creation systems

Stable Audio

neural tape loop /unloop

Suno

MusicFX DJ

RAVE

musical instruments

DAW software

modular

saxophone

delays

live coding language

violin

looper

drum guitar

music box

step sequencer

piano

granular synthesizer

musical interaction

composing

fastest human reflex (100ms)

human lifespan

live coding

DJing

playing a traditional instrument

time threshold of pitch perception (45 ms - 15ms)

live looping

scale

divisions of form, phrase structures, etc.

sound particles that make up sound objects

duration of a musical movement/ genre

duration of a single composition

a 'note'; A single sound object

years ... 15min 2min 10s 1s 100ms 10ms 1ms 100μs ...

more "offline" *more "real-time"*

Figure 4.1: Several generative music co-creation systems, musical instruments, and musical interactions, organized into their respective time scales of music. Chapter 5's contribution, the neural tape loop, generates *meso*-scale musical structures. *macro*-scale co-creation systems (Suno) are too *detached* from the music – they afford casual musicmaking experiences not suitable for a sound artist or instrumentalist. *Sound object*-scale co-creation systems (RAVE) offer an immediate sound-producing interaction, similar to traditional acoustic instruments. A *meso*-scale co-creation system (neural tape loop) sits in between: it leverages two-stage generative models to generate longer musical structures (up to 10 seconds) while remaining interactive enough to be used in live performance and art installations.

musical interaction, examining the level of involvement of a musician or sound artist in a particular interaction.

Now, let us consider generative AI models (and how we interact with them) in the context of Road's time scales of music. RAVE models are relatively *light* models with a short temporal context, typically trained on a unitimbral distribution of sounds for best effect (e.g. "human voices", "violins", "recordings of water", etc.) on a $\tilde{3}$ second context. From an interaction standpoint, RAVE instruments are mostly capable of immediate sound-producing interactions (by directly manipulating latent dimensions with a controller, e.g. Stacco [19] and semilla [18]), or via timbre transfer.

Using Road's organization of the time scales of music [150], I would classify that RAVE's generative properties primarily operate at the *sound object* level – the basic unit of musical structure. The *sound object* time scale generalizes " the traditional concept of note to include complex and mutating sound events on a time scale ranging from a fraction of a second to several seconds" [150]. RAVE models (without prior, which is the most commonly used setup in nimes) are not generative at the *meso* level: they are not able to generate coherent sequences of sound objects by themselves, leaving the arrangement and transfiguration of these mesostructures up to the performer and their controller (in the case of Stacco and Semilla) or their instrument (in the case RAVE timbre-transfer systems).

Commercial AI music co-creation systems, on the other hand, involve the musicker at larger time scales. Products like Suno, Udio and Stable Audio operate at the *macro* level, where entire musical compositions are made with little temporal intervention by the sound artist. The interaction is usually in form of a text-prompt that is used to generate a full-form track (roughly 2-15min). At this level, the musicmaking interaction is at a *curatorial* level, where the majority of the interaction time is spent listening to generated tracks and curating them in the form of playlists or DJ sets. Full-form music generation systems have fostered a new community of *Prompt Jockeys*: DJs that mix AI music generated live instead of "real", previously existing music [8]. Another example is

---

[8]see `https://youtu.be/_fpnAHoRSqU`

Google's MusicFX DJ, which allows one to interact at longer durations of the meso level as well as the macro level, allowing one to create music by typing and mixing short text-to-music prompts.

In the next chapter, I'll discuss **the neural tape loop**, a generative musical instrument based on masked acoustic token modeling, a two-stage generative modelling approach for sound. Unlike RAVE, a generative musical instrument based on a masked acoustic token modeling cannot afford real-time sound-producing interactions due to its non-causal nature, nor can it produce full-form music tracks with a single premeditated command (like Suno): instead, its primary time scale of interest for musical interaction lies at the *meso* level, creating local rhythmic patterns of sound objects, pitch and timbre melodies, establishing themes, variations and textural interplay. This places the neural tape loop on the same time scale of interaction as sequencers, loopers, modular synthesizers, delays, etc. According to Roads, the *meso* level is extremely important in composition, for it is at the *meso* level where the "sequences, combinations and transmutations that constitute musical ideas unfold".

CHAPTER 5

THE NEURAL TAPE LOOP
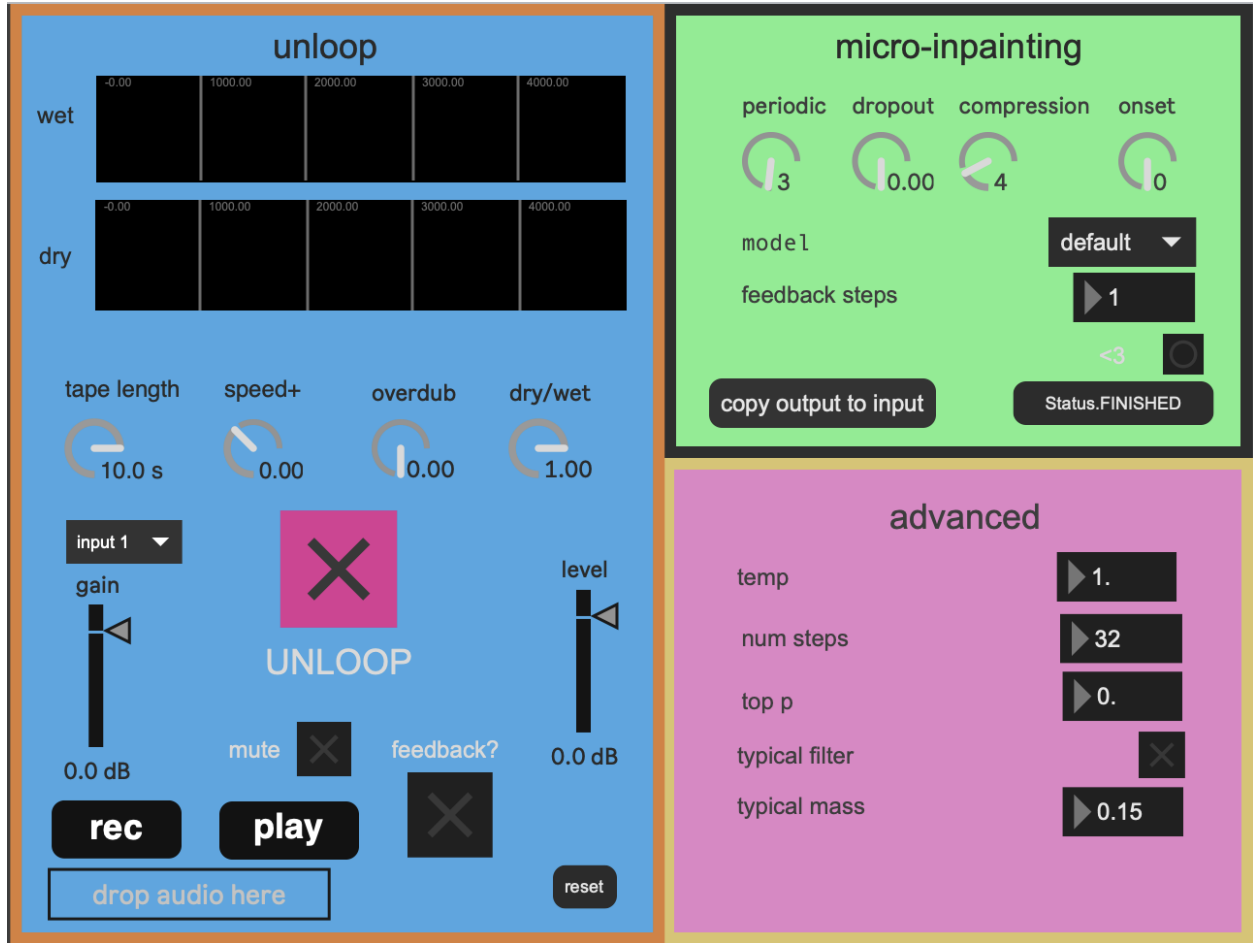
Figure 5.1: *unloop*, a digital musical instrument built with a neural tape loop. *unloop* equips a digital looper with a masked acoustic token model so that the loop "never repeats itself": as the recorded loop plays back over and over again, the original contents of the recorded loop are transformed to reflect a generative model's sound palette. The transformation can be perceptible at both the sound-object scale (*timbre transfer*) or at the *meso* scale (*rhythm/phrase structure transfer*), depending on the micro-inpainting controls used.

Generative models are not neutral entities existing in a vacuum. They are shaped by the data they are trained on, the infrastructures that build them, and the ideologies that guide their design and use. But they do not have to be instruments of a commodified pastiche. If we, as makers of

generative musical systems and instruments, engage with the systems we make critically, aesthetically, and performatively, we can make generative models that engender a new musical tradition – one that values humanness, experimentation, embodiment, and expressiveness.

This chapter introduces the **neural tape loop**, a co-creative generative musical meta-instrument based on masked acoustic token models (MATM) and their musical affordances. Through practice-based research, I show that the neural tape loop facilitates novel musicmaking techniques and interactions that can be deployed in different scenarios belonging to experimental music and the sound arts. Specifically, this chapter presents four original works – a mono fixed media composition, a comprovisation (composed improvisation [169]), a multichannel fixed media composition, and a quadraphonic interactive sound installation – each exploring and introducing a musical technique enabled by masked acoustic token models.

I propose three new techniques for playing a masked acoustic token model: **micro-inpainting**, **fed back iterative regeneration (theseus sampling)**, and **generative time stretching**, each of which enable new musical interactions. I emphasize the importance of enabling sound artists to use their own sound collections when working with generative models, and discuss my use of an existing *sound palette fine-tuning* technique. I employ these techniques in my own original compositions, performances and sound installations, as well as contextualize them within the experimental and computer music traditions. I show how these techniques, together, allow one to use *the voice as the interface* for the generative co-creation system, allowing one to create morphs between one's voice and any desired sound palette (Section 5.4.3).

The work presented here aims to extend the lineage and discourse of both experimental music and generative audio modelling by situating the new wave of two-stage generative "AI" models within a broader musical ecology - in this case, one shaped by historical practices of sonic hauntology, tape manipulation, looping, and feedback, and sustained by communities invested in experimental sound, improvisation, and process-based music.

The neural tape loop is a co-creative musical meta-instrument [170]. The neural tape loop expands on the work of VampNet [16] and reframes the masked acoustic token modeling (MATM)

system as a meta-instrument [170] from which musical instruments, sound installations, audio processing interfaces, and other musicking [40] interactions can be built.

In the neural tape loop, I introduce new techniques that leverage the affordances of a masked acoustic token modelling system to transform an input sound from its input distribution (e.g. 'voice') to the model's trained distribution (e.g. 'machines'). The model may be fine-tuned to match a sound artist's desired sound distribution by letting the artist provide a relatively small collection of sounds (in the order of 5 minutes to 12 hours), referred to as a *sound palette*.

By employing different masking techniques (micro-inpainting, theseus sampling, generative time-stretching, see Section 5.2), a sound artist may perform these transformations at perceptually different time scales (*meso* and *sound-object*, see section 4.7) [150]. For example, with a neural tape loop, one could employ a transformation at *sound-object* level, resulting in a timbre transfer-like transformation. One could also transform at the *meso* level, creating structural changes to the input sound by modifying a rhythm or introducing/omitting sound objects from the acoustic token sequence.

While I focus on my work with VampNet, these techniques can be applied to any generative model capable of performing masked acoustic token modeling or discrete diffusion.

I frame the neural tape loop as a *meta-instrument*: an instrument from which other instruments can be built [170]. While several interfaces have been built for the neural tape loop (see the looper "unloop", discussed in section 5.4.2), the following section discusses techniques applicable to the masked acoustic token modeling system regardless of its front-facing interface. This gives us the flexibility to create different instruments from the neural tape loop – like loopers (section 5.4.2), DAW-extensions (section 5.4.3), interactive sound installations (section 5.4.4), and even samplers (future work!).

5.1 Sound palette fine-tuning

All of the acoustic token manipulation techniques introduced in this chapter are most effective when an artist has the power to fine-tune their generative model to their own *sound palette*: a

bespoke collection of sound recordings for the neural tape loop to model.

While magnetic tape (and traditional digital audio buffers) contain explicitly defined sounds in them, an acoustic token buffer can represent a learned generative distribution of sounds which can then materialize as different sound structures according to the model's learned distribution. New sounds can be created by processing masked acoustic token prompts created from input sound queries by performing inference with a masked acoustic token model like VampNet[16]. By letting individual sound artists bring their own **sound palette**, we can give the sound artist a great deal of expressive power over the model.

Enabling artists to bring their own sound palette to a pretrained generative model like VampNet encourages a small data mindset (which enables greater human influence in a generative AI creative context [171]) while leveraging the expressive power of two-stage generative models.

From a technical standpoint, I enable sound palette fine-tuning with a masked acoustic token model by employing LoRA [102] fine-tuning on the original VampNet model. This makes fine-tuning efficient: one can fine-tune a VampNet to follow a custom sound palette on a single consumer GPU. Fine-tuning is completed typically within a day – with most of the learning done within the first 2 or 3 hours of fine-tuning, making fine-tuning a model to multiple sound palettes for a single musical project feasible even when only a single GPU is available.

The size of the target sound palette can vary greatly, from a 5-10 mins (fine-tuned on one or two songs, or a couple of individual recordings) to 2-12 hours of sounds with a specific style or texture. Anecdotally, I've found that models trained on very small sound palettes (e.g., less than 1 hour of audio) begin to exhibit "overfit" model behavior, by always insisting upon generating this training material regardless of the input given. This can make it more difficult to perform timbre and structure transformations on an arbitrary input buffer. However, with small sound palettes, the model can function as a "generative sampler" of the small sound palette – creating a sequence of mesostructures from the small sound palette, reconfigured together in non-linear ways (like a generative infinite jukebox [1]).

---

[1] https://eternalboxmirror.xyz/jukebox_index.html

Sound palette fine-tuning makes personalizing large two-stage generative models accessible to computer musicians who may have access to a single consumer GPU (like a gaming PC). I've included a tool allowing computer-savvy sound artists (i.e., people with basic command line skills like copying/moving files and launching a python script) fine-tune their own VampNet models in the open source VampNet repo [2].

5.2 Token Manipulation Techniques

Here, I detail several techniques for creating sonic material and making music with a neural tape loop, primarily based on acoustic token manipulation, an affordance unique to masked acoustic token models (MATM).

When employed together with sound palette fine-tuning (Section 5.1), these techniques allow for powerful and embodied interactions for transforming and manipulating small to medium-sized collections of sound material, like shaping sound material with one's voice (Section 5.4.3), or spatializing a monophonic texture into arbitrarily many channels with a generative effect (Section 5.4.4).

To the best of my knowledge, no other work has introduced these token manipulation techniques. In addition, no one has explored the affordances and aesthetic implications of masked acoustic token models within the context of a larger creative practice (Section 5.4).

5.2.1   Micro-inpainting

A neural tape loop can be built with any generative model capable of re-generating sound from a masked (or corrupted, in the case of diffusion models) input. However, in order to perform diverse kinds of sound transformations resembling timbre transfer, one should be able to configure the input's *unmasked* sections to be very short (a single acoustic token's approximate duration, around 20 ms). When contiguous unmasked sections are very short (e.g., 1 token wide, a "sparse" prompt), the generative process can be thought of as a form of **micro-inpainting** – inpainting at the

---

[2]https://github.com/hugofloresgarcia/vampnet
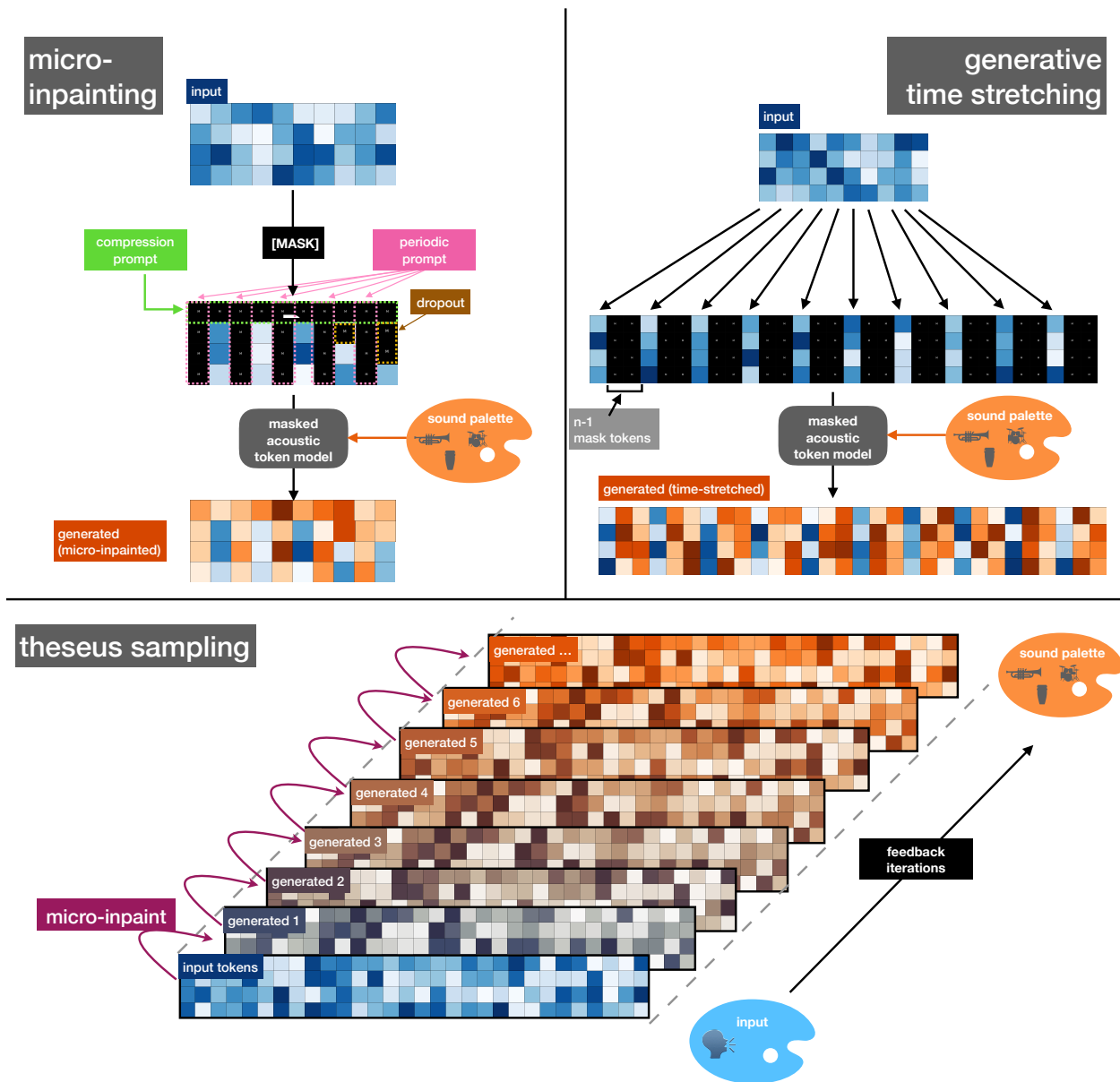
Figure 5.2: Token manipulation-based neural tape loop techniques. (top left) micro-inpainting (top right) generative time-stretching (bottom) theseus sampling.

microsound[3] [150] level – where the conditioning given to the generative model consists of short bursts of sound particles that contain only partial information about a sound's structure, timbre, pitch, etc.

Micro-inpainting allows the generative model to reconstruct an approximation of the missing information by leveraging its learned training distribution, effectively treating the masked portions as a blank canvas upon which new sonic features and structures can emerge. The total percent of masked tokens can be 50-95% of the total acoustic tokens if a meaningful variation of the sound is to be made. The masked acoustic token model thus synthesizes sounds informed by both the short unmasked microsound bursts used as conditioning as well as the patterns present in the model's learned distribution.

In order to perform micro-inpainting, we must carefully craft a mask for our input buffer that both contains the appropriate amount of conditioning information from the input audio file while allowing enough "masked space" for the model to generate. I use techniques based on the prompting techniques described in VampNet [16]. Because the resulting masks have many more masked tokens than unmasked tokens, these masking techniques begin by fully masking the entire acoustic token buffer, then selectively *un*masking tokens according the given criterion. Here, I briefly re-introduce these *un*masking techniques, as well as further elaborate on how different configurations of these masks can have different perceptual (and thus musical) effects on an input buffer of acoustic tokens.

- **Periodic Prompting (Vertical masking):** fully masks all tokens, then periodically unmasks every pth timesteps. Larger values of p leave larger masked portions, allowing the model more room for generation. p can be any value between 1 and T, where T is the total number of timesteps in the input acoustic token sequence. For a well-trained model (i.e., a model fine-tuned on a sufficiently large sound palette, anecdotally 1-2 hours for a unitimbral distribution), values of $p = 1$ and $p = 2$ produce perfect and approximate reconstructions of the input audio, respectively. Values of $p \in [3, 5]$ produce a transformation at the *sound-object*

---

[3]refer to section 4.7 for an overview of the time scales of music.

[150] time scale (i.e., **timbre transfer**), morphing the perceptual identity of individual sound objects towards the sound distribution of the trained model. With values of $p \in [7, 15]$, the model begins to exhibit generative behaviors at the *meso* [150] time scale, introducing new sound objects into the acoustic token buffer and altering the rhythmic structure. I refer to this perceptual process as **structure transfer**, since it results in a transformation of both timbre and rhythmic structure, altering the sonic mesostructures of the original input.

- **Dropout masking**: Like vertical unmasking, but timesteps are masked randomly instead of periodically, which can be useful for leaving larger chunks of space fully masked, where the model can introduce new sound events. Dropout unmasking can be parametrized with parameter $d \in \mathbb{R}^{[0,1]}$, indicating the probablility that any given token in the sequence should be masked.

- **Compression Prompting (Horizontal masking):** fully masks all tokens above the nth specified codebook level, where $n \in \{1, N\}$, where N is the number of codebooks used by the VampNet tokenizer model. Lower values of n completely mask more codebook levels, resulting in a removal of timbre at medium levels (roughly $3 < n < 5$ for VampNet), and a complete removal of timbre and pitch at low levels ($n < 3$).

- **Onset-based prompting**: like VampNet [16], we can also leverage information about the input audio to construct sonically-informed masks. Since VampNet's beat-based prompting is unsuitable for natural sounds, voice prompts, or free-tempo sound gestures (since it relies on a beat tracker), this work instead proposes to leverage sound event onsets predicted by a onset tracker, like the one provided by `librosa`[172]. Once all of timesteps which contain onsets have been identified, all of the tokens located at timesteps with a predicted onset (as well as the surrounding ones, dictated by an `onset mask width` parameter) will be *un*masked, ensuring that all of the predicted note onsets remain consistent in the generated output.

Micro-inpainting is the primary *performance-time* technique used to play with a neural tape loop: a sound loop can be recorded, micro-inpainted, and played back during a live performance

or sound installation, as shown in Section 5.4. All of the other techniques that follow (*Theseus sampling* and *Generative time stretching* make use of micro-inpainting.

5.2.2 Theseus sampling

Micro-inpainting is capable of performing *sound-object* and *meso* level transformations from the input sound to the model's sound palette. However, when very sparse masks with periodic prompts are used (for example, $p > 13$), the output sound may be *too different* from the original sound in terms of mesostructure, which can lead to the listener feeling as if there is a discontinuity, or a lack of correspondence between the input and output sound.

I propose a method called *Theseus Sampling*, which performs *sound-object* and *meso* level transformations by applying micro-inpainting to an acoustic buffer multiple times in an iterative, feedback manner. With theseus sampling, one can create larger *meso* level transformations by applying smaller periodic prompts (roughly $p < 13$) over multiple feedback steps. By gradually replacing segments of an input signal across multiple passes, Theseus Sampling allows the model to retain the broad temporal shape of a sound while continuously rewriting its local content. This allows the sound artist to craft transformations that range from subtle variations to radical reconfigurations. This iterative process where small parts of a whole are all slowly replaced is reminiscent of the "Ship of Theseus" paradox, hence the name "theseus sampling".

While Theseus Sampling seems conceptually similar to a discrete diffusion [74] or an iterative parallel decoding process [66], **each step of a theseus sampling process is a fully-formed output produced by a masked acoustic token model**. Unlike the intermediate outputs in an iterative parallel decoding scheme, each output in a theseus sampling process may be listened to and used as sonic material.

Theseus sampling amplifies the effects of micro-inpainting each time the output of the MATM is fed back to the input. If done with smaller periodic prompts ($p < 5$), one can create more intense timbre transfer effects than if done via a single pass of micro-inpainting. A notable example of this

technique is my open source music processing tool nesquik [4], which uses theseus sampling with a model fine-tuned on NES music to transform any instrumental music into an "8-bit" chiptune rendition of itself.

If a theseus sampling process is repeated indefinitely for a single input, the sequence of all of the resulting outputs of the model represents a single gradual sound transformation process, where any input sound can be slowly *dematerialized* from its original perceptual source into a texture defined by the model's sound palette. In section 5.4.4, I discuss *token telephone*, a sound installation that exposes the resonant modes of a neural tape loop's training distribution by revealing a theseus sampling feedback process in front of the participant.

5.2.3 Generative time stretching

MATMs afford us to time-stretch a piece of audio with an "opinionated", generative behavior which both stretches the durations sound objects in a sequence while changing the timbral properties of these sound objects according to the model's sound palette. This time-stretching technique, especially when applied at extreme time-stretching factors of n (like 5, 8, 10, 15, 20), begins to exhibit generative behaviors: the training material of the model begins to emerge from the masked spaces in between the original sound events. The aesthetic result is not simply a slower version of the original sound. Instead, the model begins to generate new sound events in the blank spaces introduced by the masked tokens, changing the identity of the input sound from the source audio to the model's sound palette, all while preserving the grand rhythmic structure of the stretched audio.

To time-stretch a piece of audio with a neural tape loop by an integer factor n, we can **insert** $n - 1$ mask tokens in between every individual token of an input token buffer. For example, to time-stretch a sound by a factor $n = 3$, we must insert $3 - 1 = 2$ mask tokens in between every token in the input sequence. This lengthens the input token sequence to $n - 1$ times its original length, effectively decoding the original acoustic tokens at a rate given by $1/n$.

Note that *generative time stretching* is different from *micro-inpainting* in that *generative time*

---

[4] https://hugofloresgarcia.art/nesquik

stretching **inserts** new mask tokens in between the input tokens, while *micro-inpainting* **replaces** some of the input tokens with mask tokens. *Generative time stretching* effectively lengthens the token sequence (thus stretching the generated audio). *Micro-inpainting* simply transforms the token sequence – the resulting token sequence is the same length as the input.

Since this time-stretching techniques exhibits its most interesting behaviors at very large time-stretching factors ($n > 3$), generative time-stretching is best suited as an "offline" technique for fixed-media compositions, like my original composition for voice and VampNet called *world of mouth*. See Section 5.4.3 for a discussion on using generative time-stretching in a fixed-media composition.

5.3   Interface: *unloop*

unloop is the primary live performance interface built on top of the neural tape loop. *unloop* places a masked acoustic token model in a live looping digital instrument, equipping the looper with the ability to create generative transformations of the loop as the loop repeats. While looping, a user can modify the parameters used for performing micro-inpainting and theseus sampling on an input sound, and can thus create different kinds of generative transformations to the sound, from a timbre transfer to mesostructure transfer to a full-on unconditional generation. The first two pieces described in Section 5.4 make use of *unloop*.

Figure 5.1 shows the *unloop* user interface (implemented as a Max/MSP patch). The left panel contains basic looper controls (e.g. loop length, speed, overdub) as well as a toggle (UNLOOP) to begin the transformation process. Unlike a traditional looper, which contains a single buffer to record sound into, *unloop* has two buffers: a *wet* buffer and a *dry* buffer. "Real" audio (like a voice gesture or input loop) is recorded to the *dry* buffer, which is used as input for the neural tape loop. Audio generated by the neural tape loop is placed in the *wet* buffer. A *dry/wet* parameter allows the user to mix the original and generated audio together.

With the (UNLOOP) toggle enabled, the contents of the current buffer are immediately sent to a gradio API server equipped with VampNet, which performs a micro-inpainting transformation

on the input buffer. The generated buffer is then sent back to the Max patch, which cues the generated buffer to replace the "wet" buffer once the current loop reaches it's end. The micro-inpainting controls (Figure 5.1, top right) allow a user to modify the mask-building parameters described in section 5.2.1 (micro-inpainting). A set of advanced controls (bottom right) modify the model's sampling process, adjusting aspects like each generations's randomness (temperature, typical filtering [173], top p). Additionally, one can enable *theseus sampling* mode by enabling the `feedback` toggle in the left panel of the *unloop* interface. With the `feedback` toggle on, the contents of the *wet* buffer are used as input to the model (instead of the *dry* buffer). This means that the model will be recursively feeding into itself, as described by the *theseus sampling* process in Section 5.2.2.

While working with the *unloop* interface for different occassions/performances/compositions, an issue that kept reoccurring was the interface's lack of a mask visualization tool. Being unable to visualize the input mask makes it hard to have a mental model of what the current mask looks like, especially when mixing the masking parameters described in Section 5.2.1. Though this was a reoccurring issue in most of the creative works described below, I did not formalize this thought until a discussion with other artistic collaborators (Weilu Ge and Nithya Shikarpur). See a proposed design guideline in section 5.4.2

5.4 Creative Works

5.4.1 living // dreaming

living // dreaming is my first musical work with a generative model, and the only unpublished piece referenced in this chapter. This piece deserves mention because it was the first time I attempted a creative project with the neural tape loop, and it led to the development of the *theseus sampling* technique. A recording of *living // dreaming* is available on YouTube [5].

The *theseus sampling* technique emerged as a result of a technical implementation bug while working on an interface for the creative work that would later become *living // dreaming*. While

---

[5] https://www.youtube.com/watch?v=KnBJIgaPZCk
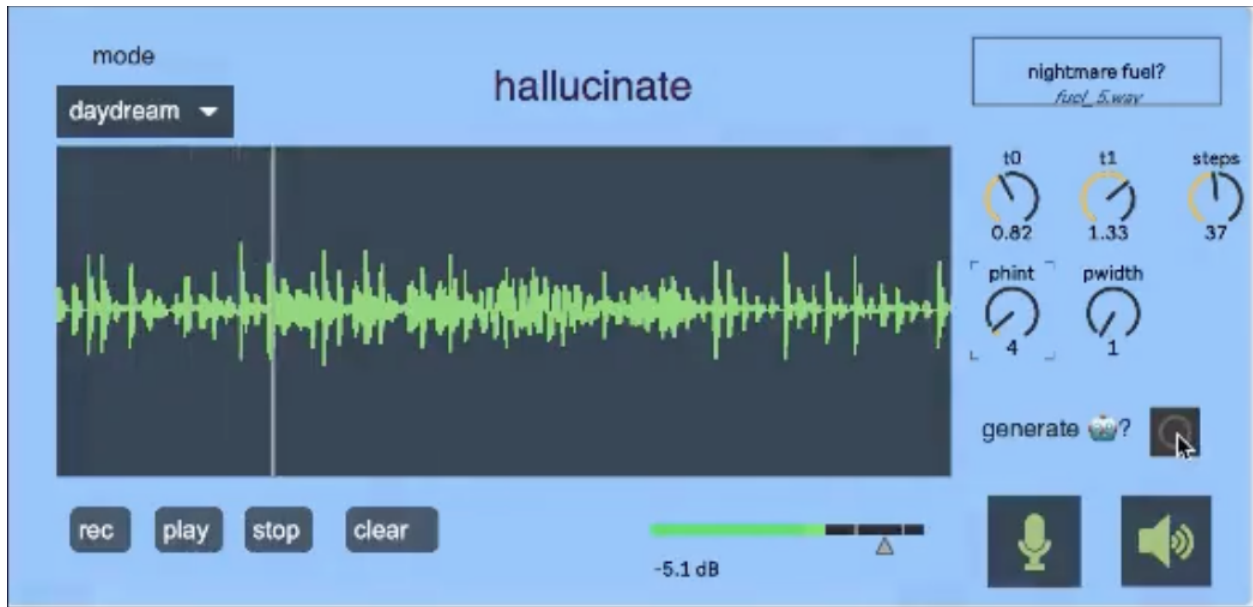
Figure 5.3: Early version of *unloop* used for the composition *living // dreaming*

working on a live looping interface for VampNet, I accidentally set up the looper so that the generated audio would overwrite the original input audio file. This meant that each subsequent model generation would use the previous generation as input, enacting a recursive sound transformation process with VampNet. Though I originally dismissed it as a bug that would only lead to sounds that were "more distorted than they had to be", I was surprised to find out that, after a certain number of these recursive transformations, larger structural changes became perceivable in the generated loop.

At the time, I was exploring this recursive transformation process with a "debugging" audio sample – a scratch piece of audio used to debug a musicmaking interface. This "debugging" audio sample was a short chaotic loop of drum samples randomly sequenced in a 32nd note grid. As the recursive transformation process ensued, I began to observe the longer-term structural changes that happened to the original drum loop – the drum loop went from being chaotic to having a more "traditional" sequence of drum hits. Through more feedback transformations, the drums-only loop then began to spew out eerie vocal textures and short bursts of harmonic and melodic instruments, like synthesizers and electric pianos.

When this recursive transformation process was left to its own devices, I observed that the

89

original chaotic drum beat contained in the neural tape loop would slowly turn into a quiet, passive texture of a sustained piano chord – what I imagined would be a "low-entropy" point in the generative model's learned distribution. I became quite interested in this recursive, flattening process that happened when the output of a generative model was fed back into itself, and so I further developed *theseus sampling* as a technique for sound transformation with VampNet. *theseus sampling* then became instrumental in making my other neural network-based creative works, like *world of mouth* and *token telephone*.

What was originally an implementation error while working on a musical composition later became a new technique for playing with a neural tape loop. After discovering the *theseus sampling* phenomenon, I modified the user interface to make it possible to control whether to apply VampNet in *feedforward* ("*living*", i.e., non-recursive micro-inpainting) or *feedback* ("*dreaming*", i.e., recursive micro-inpainting or *theseus sampling*). This allowed me to take an input sound (like a chaotic drum beat, i.e. *nightmare fuel*) and playfully transform it in both feedforward (*living*) and feedback (*dreaming*) modes – creating superficial variations as well as "washing out" the original sound into the generative model's resonant modes.
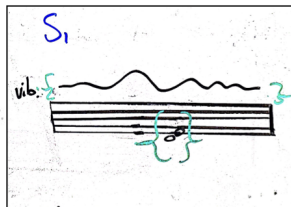
5.4.2   confluyo yo

confluyo yo (*I converge*) is a comprovisation (composed improvisation [169]) for saxophone and *unloop*, a live looping interface built with a neural tape loop. In *confluyo yo*, a saxophonist performs short modular motifs called "seeds", which are looped and transformed live using a neural tape loop system. Through repeated micro-inpainting transformations, the saxophone's timbre and rhythmic structure dematerialize (i.e., lose their semantic meaning), slowly converging into textures derived from generative models of field recordings of Central American birds and industrial machinery.

Confluyo yo was recorded in 2023 in collaboration with Honduran saxophonist Michael Pineda (saxophone), and performed at the ISMIR 2023 conference by myself (unloop/VampNet) and Bryan Pardo (saxophone). The studio recording of *confluyo yo* (recorded with Michael Pineda)
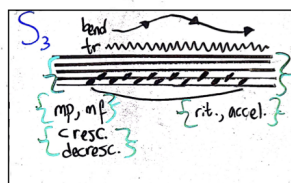
Figure 5.4: Saxophone score for confluyo yo. This score outlines 5 seed patterns (S_1 through S_5) which the player can use as initial material for the structure transfer process. In a performance of confluyo yo, a performer can play any these seed patterns into a neural tape loop, which is used to enact a gradual timbre and structure transfer process from the original saxophone gestures to two generative models trained on central american birds and industrial machines, respectively.

is available on YouTube [6].

Figure 5.4 shows the original saxophone score for *confluyo yo*. The improvisation is structured via *seeds*, or loose, modular themes that are used as source material for a neural tape loop. Seeds can be chained together into a *sequence*, used as the structural foundation for the improvisation. Seeds S_1, S_4 and S_5 are meant to evoke machine-like sounds from the saxophone, evoking the sounds of bells, engines, and typewriters from the model trained on the "machines" sound palette, while seeds S_2 and S_3 are meant to evoke birdlike sounds. In a performance of *confluyo yo*, the saxophonist performs one of the seeds, which is recorded into an unloop, which subsequently triggers a cascade of transformations of the original seed. *unloop*, a live looper equipped with a neural tape loop, creates transformed versions of the seeds, morphing the sound objects and mesostructures in the original recording to resemble the model's learned representation of Central American birds and industrial machines.

on the role of the operator

During a performance, *unloop* is controlled by the *operator*: a live performer manipulating both the looper instrument and its underlying generative tape loop. *unloop* inherits the affordances of live looping instruments (like the Tascam PortaStudio or the Boss RC-505): one can selectively record material into a buffer of sound, which can then be played back in a loop as repetitive material while the instrumentalist develops new material over the repeated loop. With *unloop*, the performer controlling the looper has an additional role: to steer the generative process undergoing in the neural tape, adjusting the micro-inpainting mask used for each transformation of the recorded loop. This allows the operator to evolve the loop two ways: by recording new material into the loop, and, by transforming the recorded material with the underlying generative model. The generative model can also be used in an "unconditional" mode, which does not need any audio input, filling the buffer with randomly generated material from the model's sound palette only.

Functionally, the operator mediates the conversation between the instrumentalist and the gen-

---

[6] https://www.youtube.com/watch?v=mcjf2iKf8Nk

erative model, shaping the transformations that the sound in the neural tape loop undergoes. From an agency perspective, *unloop* is a mixed-initiative instrument – combining the premeditations and sonic interventions of the performer, operator, and generative model into a single resulting sonic structure.

While perhaps it would have been a more interesting technical challenge to devise a system capable of fully improvising with the instrumentalist by itself (i.e., forgoing the need of a separate operator), one cannot ignore the social context surrounding AI's role in music. There is widespread concern that AI models are here to "replace" musicians. Instead, the neural tape loop adds a new performer onstage, one who plays the generative model at the mesostructure level to preserve, morph, or subvert sonic material. This includes the initiative of a new human performer that can leverage the expressive power of generative neural networks onstage.

Hauntology and embodiment in confluyo yo

From the perspective of sonic hauntology (see the section on Experimental AI music 4.5), *confluyo yo* is a "summoning" piece, invoking the sonic spectres of both natural and machine-made sounds through a performative ritual of imitation. As the generative model transforms the saxophonist's gestures, it imposes its own temporal and timbral structure on the input sound. This transformation process becomes a form of material agency – the model "plays back" with its learned distribution, surfacing sonic patterns that reflect both the input and the underlying sonic corpus. Traces of the saxophonist's gestures persist, though they become increasingly spectral, as if haunted by a hidden musical entity encoded in the model.

The neural tape loop enabled a speculative form of sonic embodiment, where a performer can construct and inhabit new structures of sound objects through a gestural interaction with the generative model. Rather than controlling an unconditional process or triggering pre-determined sonic responses, the performer must develop a musical language that resonates with the model's learned sound palette. This can be thought of as a dialogical engagement with an imagined sonic identity, an uncanny sonic counterpart that transcends traditional notions of instrumental control

into an evolving, "haunted" sound ecology.

In an informal conversation taking a break in between recording sessions, Pineda described his "performance philosophy" at the moment as an attempt to "become" the model's imagined sonic identity. He reflects:

> "With the horn, dropping in, coming from nothing to something – what does that sound like, and how can I become that sound? How can I be *machine*? How can I be this amorphous thing?"

Weilu Ge's Cat in loop x Catinblack

Though not my own creative work, another composition worth briefly mentioning for its use of a neural tape loop is composer Weilu Ge's *Cat in loop x Catinblack* (2025). The following are Ge's program notes from a performance of *Cat in loop x Catinblack*:

> Cat in loop x Catinblack is a new work for string trio and generative Al music system, exploring the collective imagination of cat musicality. Through an interactive and recursive process of listening, playing, and performing, we collaboratively brought an imaginative CAT to life enacting a posthuman subject-in-progress: a relational, embedded, and embodied "we" that moves fluidly across environmental, cognitive, technological, visual, and sonic realities. Vampnet, developed by Hugo Flores Garcia, is an AI model trained to re-write parts of an audio input, i.e. "vamp" on it. Fine-tuned on cat-inspired audio data - mixing real, synthetic and imagined cat gesture sounds, this model allows us to creatively reinterpret sounds in a cat-like manner both in expected and unexpected ways. We procedurally adapted the system with creative inputs from the catinblack ensemble and engineers during a four-session HGNM residency at Harvard. Throughout the process, a palette of sonic and gestural vocabularies was co-developed with the ensemble as the piece gradually took shape.

Similar to what Pineda and I found when working on *confluyo*, Ge's work with the *Catinblack* string trio aimed to use the neural tape loop (vampnet) as a way to mediate a collective embodiment

process. In the case of *Cat in loop*, the neural tape loop serves as a medium where the ensemble collectively imagines a "cat" through a neural tape loop. When discussing the performance and composition process retroactively, Ge referred to the composition process as the development of a common language between the performers and the generative model (which embodies the "imaginative cat").

This aim to inhabit and embody the neural tape loop's imagined sonic identity illustrates a new kind of instrumental relationship made possible by these generative models. This relationship is not one that is rooted in fine-grained direct "control" of a sound production mechanism, nor is it one that offers a detached high-level steering of a grand statistical process, but one that facilitates the embodiment of a sonic process through the development of a common sonic language.

Design Remark: Miscommunication! Dealing with Communication Mismatches with a Generative Model

In my practice with unloop, I have observed that this sense of embodiment can be undermined or even turn into a mild frustration when the model doesn't respond to the artist's intended gestures (e.g., when a user makes a *meow* sound into a model trained on cat vocalizations and the output turns into something other than a cat). We experienced this when working in *confluyo yo*, and Ge experienced this when working on *Catinloop* as well.

A musical approach to deal with this issue is to spend time developing a common sonic language that is more or less guaranteed to work in a live scenario. For *confluyo*, I worked out a set of motifs (seeds) for the performer (Figure 5.4) that were likely to *evoke* certain specific sounds of birds and machines from the model. To ensure a successful live performance, Ge spent a session exploring the boundaries of the model with the performers for *Cat in loop*.

A technical approach to this issue could be to leverage explicit control signals instead of masked token prompts, like Chapter 3's Sketch2Sound [17], or The Rhythm In Anything (TRIA) [132] – a work led by my labmate and collaborator Patrick O'Reilly.

Yet, adding explicit conditioning signals may lose the surprise that makes the neural tape loop

feel like a larger-than-life process. In my preliminary experiments attempting to compose music with Sketch2Sound (a voice-to-sound synthesis system that is more "controllable" than VampNet, introduced in Chapter 3), I've found that a model that "always meows" when you ask it to can feel sterile, as this erodes the model's ability to introduce surprise through *welcome* misinterpretations. Even after working on Sketch2Sound, I've found myself opting to compose with VampNet instead. An interesting future experiment could be to re-train Sketch2Sound with both control signals *and* masked acoustic token modeling capabilities, allowing me to leverage both techniques, further balancing control and surprise.

Design Remark: Mask Visualizations and Freehand Masking

The operator who played with unloop for the Weilu Ge piece (collaborator Nithya Shikarpur), along with other more "casual" *unloop* users, reported having a difficult time getting an intuition for how the micro-inpainting controls (periodic prompt, compression prompt, etc.) affect the resulting sound, and how much "generative freedom" the model has at any point in time. I realized that, for a person who was new to *unloop*, the micro-inpainting controls made it hard to mentally visualize what the resulting mask used for micro-inpainting looks like. While I had built the internal mental image of the masks while developing the VampNet code, another artist playing with *unloop* could have no way of having a mental model of the masks being used to transform sound with the model.

In order to alleviate this issue within the tight turnaround time before the performance of *Catin-loop*, we came up with a temporary solution: a system for storing different micro-inpainting presets like "small variation", "timbre transfer" or a "large stuctural variation".

A better, longer term design suggestion for people working with masked acoustic token models for musicmaking would be to build in a mask visualizer into the interface that displays the acoustic token mask (or a small section of it) as the user adjusts the controls for the micro-inpainting controls. The resulting mask could be overlaid on top of the waveform for the loop, in order to know exactly which sections of the audio buffer are getting masked out, and which ones are being kept.

Another improved approach could be to drop the micro-inpainting knobs, which borrow the

affordances of knobs and analog synthesizers, along with the notion of a "periodic" and "compression" prompts, all together. Instead, we can let the artist freely draw their own micro-inpainting masks on the input token buffer as desired, letting them precisely choose which sounds to re-generate and which ones to keep intact.

5.4.3 *world of mouth*: The Voice as the Interface // Generative Time Stretching

world of mouth is an 8-channel fixed media composition built entirely from the gestural interplay between my own voice and a neural tape loop.

In this piece, vocal gesture was the primary interface for sculpting sound objects and their phrase structures, using them as input material for a set of neural tape loops trained on Central American birds, industrial machines, percussion instruments, and other miscellaneous environmental sounds. The goal of *world of mouth* was to explore the extent to which vocal performance (through improvised gestures, rhythm, and articulation) could serve as an expressive communication mechanism between my own musical intentions and a generative sound process.

The form of *world of mouth* consists of several "sound worlds", or imagined spatial sonic ecologies. Each of these is built from a separate improvised recording of vocal gestures, which is used as the primary source of phrase structure in the composition. The structure of each world is given by the shaping vocal technique (e.g., beatboxing, tongue clicking, "chewing sounds", fluttery vocal chirps, singing, etc.) as well as the neural tape loop's underlying sound palette, which "bites back", imposing its own rhythms and mesostructures into the composed world.

world of mouth was composed with the guidance of Chris Mercer, as part of a composition class at Northwestern. *world of mouth* premiered at Experimental Sound Studio [7] February 2024 in Chicago, IL, USA during the first installment of the Chicago Creative Machines [8] series and was featured at the UNPOP multichannel listening environment at Burning Man 2024 [9].

---

[7] https://ess.org
[8] chicagocreativemachines.com
[9] https://unpopularmusic.camp/

When *sound palette fine-tuning*, *micro-inpainting*, and *theseus sampling* are combined with a sound artist's voice as input, the neural tape loop facilitates voice-to-sound transformation. With this approach, vocal gestures serve not just as sound objects but as gestures that can then be reinterpreted through a neural tape loop's learned distribution. The result is a metamorphic transformation in which vocal utterances are reshaped into new timbres, rhythms and structures that reflect the model's underlying sound palette.

My work with *world of mouth* emerged from an interest in Trevor Wishart's work *Vox 5*, which is based on creating sound *metamorphoses* between vocal sounds and non-human sounds through extended vocal techniques [4]. Wishart writes of his composition: "The primary aural focus of Vox-5 is a (super)human voice that metamorphoses into many recognizable sonic images, such as the sounds of crowds, bees, a horse, or bells" [4].

While Wishart had to meticulously select target recordings, precisely align them with a vocal utterance, and painstakingly perform manual transformations using phase vocoder software, the neural tape loop substantially simplifies this process. Instead of manually sourcing, aligning, and interpolating specific sounds, the artist only needs to provide a vocal recording and a target sound palette. The neural tape loop performs the transformation without a specific target recording or manual alignment, allowing the user to explore and refine the transformation intuitively through masked prompts.

world of mouth marks my first project exploring the research idea of using one's voice as a conditioning signal for a generative model. In retrospect, the compositional and technical experiments that gave shape to *world of mouth* led me to explore more controllable ways of synthesizing sound objects from vocal performances of those sounds. This eventually paved the way for Sketch2Sound [17] (Chapter 3), a generative modeling system for sound design capable of generating sounds from sonic imitations and text prompts.

generative time stretching

An example of the generative time stretching technique can be heard in *world of mouth*. The third section of *world of mouth* (1'35" - 3'38"), was entirely constructed from a single source recording of sounds made by clicking my tongue. Before this tongue-clicking recording was used as material for the neural tape loop, it was sped up 2x using playback-rate shifting, resulting in the fast, clicking pseudo-melody heard at 1'35".

The source excerpt of tongue-clicking noises was generatively time-stretched at different factors (3x, 5x, 10x) with different sound palettes (ones trained on "machines" and "percussion" sounds). These generative-stretched sounds were then edited, layered and mixed together in a DAW, resulting in the second section of the piece.

One may hear how the tongue clicks dematerialize into sounds that resemble hand percussion instruments like bongos and congas at 2:09, industrial machines (like a cash register at 2:33), or marimbas at 2:41. The aim was to create sounds whose perceptual identity lies between those of tongue clicks and the training material of the generative model, which have unique aesthetic qualities whose existence is only possible thanks to the affordances of masked acoustic generative models, like the "guttural woodblock rolls" that can be heard at 2:18.

Leveraging the (originally un)intended artefacts of time-stretching algorithms as a musical technique is, of course, not unprecedented in the computer music and sound design traditions: Paulstretch (created by Paul Nasca) is a time-stretching effect made for extreme time-stretching of sounds (factors of 5, 10, 20), which employs phase randomization instead of unwrapping to preserve phase alignment [174]. This technique makes it suitable for these extreme stretching factors, and introduces artifacts (instead of avoiding them) that give sounds "washed out" aesthetic qualities. This technique is popular in ambient music and film soundtracks, often used to build soundscapes and lush, slowly evolving textures.
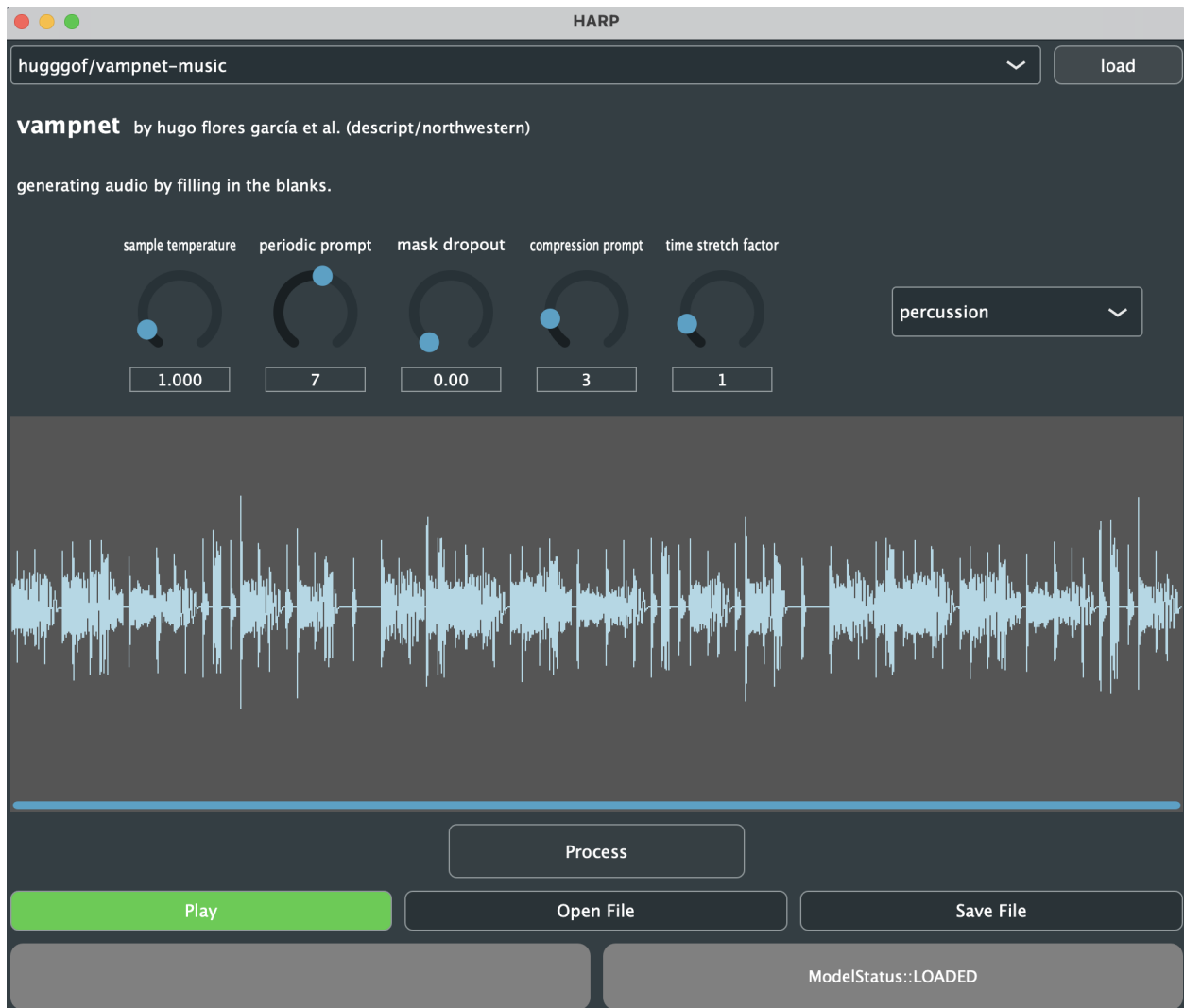
Figure 5.5: HARP user interface. HARP integrates into the DAW as an external audio editor, allowing one to process and transform tracks in the DAW with generative models (like VampNet) without having to tediously export/upload/process/download/import every audio file one would like to process.

Discussion: HARP to the rescue

world of mouth was my first fixed-media composition made with a neural tape loop in a DAW. Throughout the composition process, the generative model served as the main means of producing sonic material. I used the DAW to record vocal utterances before transforming them with the generative model, then processed these utterances with the neural tape loop. I then composed, arranged, spatialized, and mixed the transformed sonic materials into a full-form composition.

During the early stages of *world of mouth*, I used a VampNet's web demo – a Gradio web UI for processing audio with VampNet – as a means of taking my vocal utterances and transforming them with VampNet. This workflow was cumbersome – it required me to export my vocal utterances out of the DAW as separate files, manually upload them to the web, process them with VampNet, then manually download them and re-import the processed audio back into the DAW before being able to listen to the new audio in-context and work it into the mix. This became especially annoying when I wanted to audit small variations on a transformation in the context of the full composition.

While working on *world of mouth*, I was also (at the moment) the lead developer for a team working on an application called HARP, an interface for connecting researchers to music producers by connecting their deep models to an integrated DAW workflow. Before HARP, our lab had already been playing with the idea of integrating deep models into the DAW before I started working on world of mouth (see the audacitorch[10] project, a development effort that bridged small PyTorch models with the Audacity DAW, also led by me). However, world of mouth created a personal need for myself to use my own generative model in the DAW, making me the prototypical HARP user. This made the need to have a practically viable research outcome more urgent, as I wanted to something that would make composing this piece possible for me. Having my own input from a producer's perspective sped up the development and informed interfaces design choices made when working in HARP. Conversely, HARP (after several design iterations) greatly sped up my interactions with a generative model in the DAW, as HARP's DAW integration meant that I no longer had to export/upload/process/download/import over and over again.

---

[10] https://github.com/TEAMuP-dev/audacitorch

This allowed me to iterate much faster on different ideas for the piece, as I could audit different kinds of sound palettes/micro-inpainting prompts/vocal gestures without undergoing extra grunt-work, which made the composition process less frustrating, even when the model's generations failed to meet my expectations, as it required little effort to retry, generate and audit a separate idea or variation.

5.4.4 Token Telephone: Acoustic Token Feedback as a Gradual Process

Figure 5.6: Interface/instructions for *token telephone*. These were displayed on a computer monitor next to the microphone participants could use to interact and begin a new token telephone process.

Token Telephone is a quadraphonic interactive sound installation created in collaboration with sound artist Stephan Moore. In Token Telephone, participants enter a space equipped with a microphone and a quartet of generative sound neural networks, each represented by a loudspeaker. Upon vocalizing into the microphone, the participants' utterance is transformed into neural acoustic tokens and played back, initiating a game of telephone between the neural networks. Each network encodes, processes, and reconstructs the sound, distorting the original utterance into new textures guided by the network's training data. The newly reconfigured sound is then passed to the next

102

network/loudspeaker in a clockwise direction, and the process repeats. The sound produced by the fourth network is passed back to the first network in the cycle, creating a feedback loop wherein the original utterance incrementally loses all of its original characteristics and disintegrates into textures that reflect the inherent biases of the generative models in play. In time, the resonant properties of the processes are revealed in front of the participant. The name *Token Telephone* is a reference to the children's game of telephone, where a whispered message is passed from person to person, gradually mutating the original message as small mishearings accumulate. Like Token Telephone illuminates the gradual formation of hallucinations through the iterative processing and re-processing of a sound with a generative model, reflecting the biases introduced by the model's understanding of sound objects, as well as the data that was provided to it.

Token telephone is a collaboration between sound artist Stephan Moore and me. The work was exhibited at the NIME 2024 conference in Utrecht, NL. A teaser video for *token telephone* is available on YouTube [11].

Token Telephone leverages the theseus sampling (Section 5.2) technique to facilitate this feedback generative process. I use a schedule of different micro-inpainting masks to carry out the gradual sound transformation: early iterations of the theseus sampling process use micro-inpainting masks with fewer mask tokens, preserving more of the input, while later iterations increase the amount of masking, allowing the model to create larger structural metamorphoses from the input sound to the model's underlying resonances.

Conceptually, this piece is inspired by Alvin Lucier's *I Am Sitting in a Room* (1969), in which Lucier records himself speaking into a tape machine, then repeatedly plays back the recording into a room, re-recording the result onto the same strip of magnetic tape. Over time, the room's resonant frequencies become more and more prominent in the recording, until the resonances in the room become dominant, and the speech is lost in a wash of harmonic tones. Through clever tape manipulation, Lucier was able to imprint the room itself onto his original utterance, resulting in one of the most important pieces of process-based experimental music in the 20th century. The

---

[11] https://www.youtube.com/watch?v=vEaYoEgtSUo
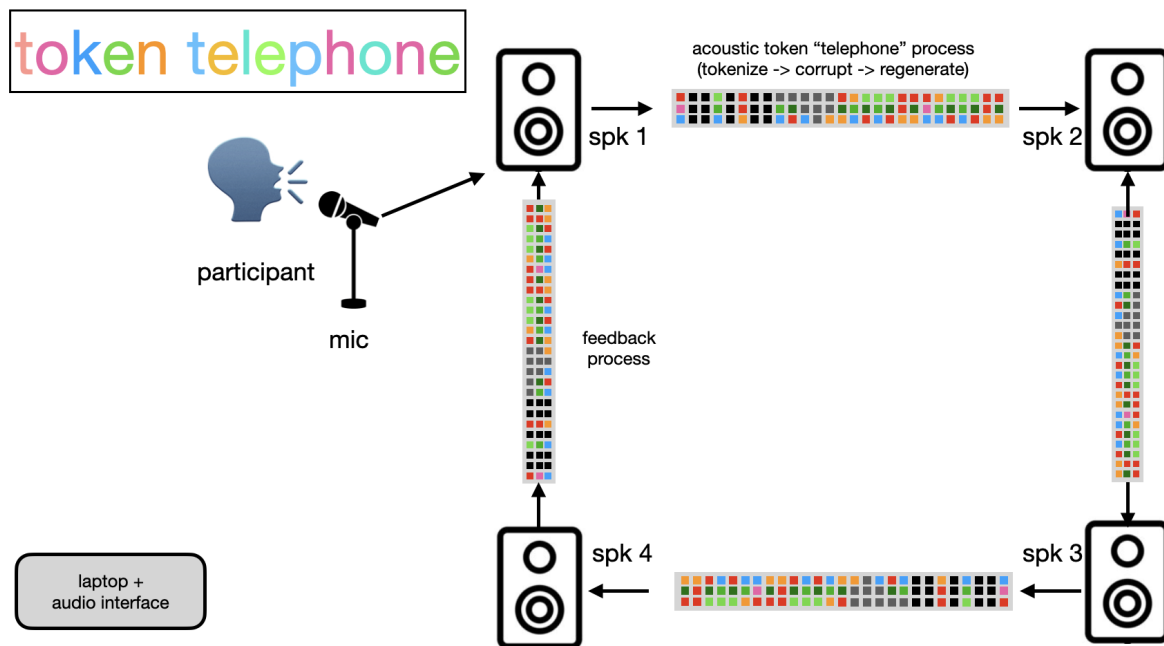
Figure 5.7: Token Telephone is a co-creative AI sound installation where participants interact with a chain of generative AI models, initiating a generative game of telephone. The installation space is circled by four neural networks, each represented by a loudspeaker. Participants make sounds into a microphone at the entrance of the installation space. Their sounds are iteratively transformed by each neural network in a feedback loop, deviating further from the original with every pass. This iterative process reveals patterns between the input and the training data of the networks, slowly morphing the rhythms and timbres of human utterances into new and unexpected sound textures.

work also bears resemblance to sonic hauntologist William Basinski's *The Disintegration Loops* (2002), where the physical decay of a repeating loop of magnetic tape becomes the core musical process.

While minimalist in structure, *Token Telephone* does not rely on external control or formal development. Instead, it reveals its form through the emergent behavior of a generative process, situated in a feedback loop with a generative agent. This recursive structure also offers a quiet critique of contemporary generative models: that these systems are not neutral, but shaped by the data they are trained on. By allowing them to "speak" for long enough, we hear what they want to say the most.

Compositional guideline: alternate realities spatialize well

A "spiritual predecessor" to *token telephone* worth mentioning here is *salad bowl*, another collaboration with Stephan Moore. The premise to *salad bowl* was similar to that of *token telephone*:

> To begin the interaction, a participant must make a sound into the microphone in front of them. The neural network then takes the participant's sound as input and destroys around 80-90% of it, then attempts to regenerate the original sound — though it knows very little about the original sound (human speech)! However, it does know a lot about natural sounds and specific musical styles. This process is repeated a number of times in a feedback loop and shown to the participant.

Stephan and I exhibited *salad bowl* at the NeurIPS 2023 Creative AI Workshop in New Orleans, LA, USA. Due to limitations encountered at the venue, we had to set up *salad bowl* as a headphones-only installation. Out of concern both for creating an appropriate stereo image for the participants, I decided to allow 2 participants to play with a separate *salad bowl* at the same time. Both (completely independent) transformation processes were then played back to both participants, panned hard left and right in the participants' headphones.

While *salad bowl* was generally positively received, I found it to be less effective at exposing the participant to the gradual transformation process of their voice than I had hoped. A number of

participants left the installation confused about the installation's purpose and message, and some had trouble hearing a pattern between the input and output. In retrospect, mixing two independent transformation processes into participants' headphones may have created an overloaded listening environment for the participant, making it harder for the participants to recognize the pattern transformation process from their voice to the model's sound palette. Before I could design another sound installation for VampNet, I had the compositional challenge of finding a way to spatialize a generative model that creates rich and noisy monophonic textures.

For my next sound installation (*token telephone*), I decided to retry the concept of an interactive voice-to-sound palette transformation process, this time without the two independent parallel streams used in *salad bowl* to create a spatial image. Instead, to spatialize the transformation process, we can play previous iterations of this same process in different speakers, moving each iteration down a speaker as new iterations come in, round robin style. These *alternate realities* (neighboring generated iterations of a *theseus sampling* transformation process) tend to sound very similar, so playing them at the same time from different speakers creates a pseudo-stereo image, creating the illusion of a single sound-object with a rich stereo image. In regions where two generative iterations are less similar to each other, this technique can create a contrasting texture on all channels, which in turn may also create an interesting spatial structure within itself. For NIME 2024, Stephan and I decided to explore this concept in a quadraphonic format, each speaker playing *alternate realities* of an infinitely-generating voice transformation process out of each respective speaker.

The quadraphonic setup created a rich spatial image for participants to experience. A quadrilateral of speakers also created an enclosed space where participants could be surrounded by the generative sound process, which made the piece feel more situated in a participant's physical space than salad bowl. Stephan and I put four chairs at the center at the room and let participants sit and listen to their sounds transform after recording into a microphone.

5.5 Conclusion

This chapter introduced the neural tape loop, a generative musical meta-instrument embedded in the lineage of experimental music, tape-based computer music, and human-AI co-creation systems. Through practice-based research, I have demonstrated that masked acoustic token models, when situated in a rich musical ecology, afford new embodied and expressive musical techniques that go beyond the typical consumer-facing use cases envisioned by commercial AI developers.

I discussed four compositions, *living // dreaming*, *confluyo yo*, *world of mouth*, and *token telephone*, each demonstrating how the neural tape loop techniques introduced above can be used in compositions, improvisations and sound installations. Rather than perfectly emulating the perceptual qualities of an existing human artist or musical style, the neural tape loop instead invites its players and participants to embody the underlying sonic material contained by a model, creating a musicmaking interaction where musical meaning emerges through gesture, transformation, and co-constructed sonic identity.

I described how being engaged in a prolonged creative practice with generative models brought about accidental discoveries of new techniques for manipulating and spatializing generative sounds, design guidelines for co-creative AI interfaces, and an interaction philosophy that led to a considerably significant contribution in the field of generative modeling for audio [17].

If you (the reader) are a musician encountering generative modelling systems in depth for the first time, I hope this chapter has shown that generative models can allow for new, unique and compelling ways of interacting with sound. If you (the reader) are a researcher working on a co-creative AI musicmaking system, I hope this work inspires you to not just evaluate your systems by benchmarking output quality, but to actively engage in a lived musical practice with your system.

Not all generative models are made to function as a commercial endless spigot of "human-sounding" AI generated slop – which we've already seen invasively and slowly seep into our streaming services [175], alienating our musical communities and weakening the connections between human musicians and human listeners by replacing musicians with an abundance of

(de)personalized, algorithmically generated music.

CHAPTER 6

EN CONCLUSIÓN

This dissertation described work I completed from 2023-2025 across the fields of machine learning, audio signal processing, and computer music. Throughout this time, I was able to design and build generative modeling systems that allowed for more controllable and expressive musicmaking interactions than existing two-stage generative modeling systems. I also composed, performed, and exhibited new original creative works built primarily using my proposed generative music-making systems.

Chapter 2 introduced VampNet, a masked acoustic token modeling approach for generating sound. Before VampNet, acoustic token generation methods relied on an autoregressive modeling approach, which required one full inference step through the model per timestep, making these systems slow and unwieldy to use in interactive applications; a user would have to spend a couple of minutes waiting for the model to process. VampNet sped this process up by an order of magnitude, as VampNet can sample 500-800 acoustic tokens in as little as 36 steps. VampNet, paired with handcrafted optimizations like `torch.compile`, can generate 10s of audio in less than 2s of inference time on a single GPU, making it possible to use a large generative model in a live performance with a live instrumental performer (Section 5.4.2).

Additionally, autoregressive modeling approaches were (by design) only made for next-token prediction: performing generative sound transformation operations were off the table: things like creating a variation of an existing sound, transforming one's voice to another sound, or transforming the structural properties of one sound to another were off the table. Using the techniques I introduced in this dissertation (Chapters 2 and 5), we can now use two-stage generative models for creating variations of a sound, transforming the structure of a sound, transforming one's voice into another sound, etc.

Several technical research works followed VampNet, adding the ability to generative individual

109

stems conditioned on other stems [70], making generation faster by modeling masked spectrograms instead of acoustic tokens [131], conditioned on text prompts [57], beatboxing [132], among others.

VampNet made it possible for large generative models like acoustic token generation systems to be used in live interactive musicking formats like music performance and interactive sound art installations, as shown in Sections 5.4.2 and 5.4.4, respectively. Before VampNet, generative models suitable for live performance (like RAVE and ddsp) were restricted to generating short-context, sound-object level (under 3s) sounds. Thus, these existing models are used either for *timbre transfer* or as an immediate *sound-producing* mechanism (akin to a traditional instrument). On the other hand, VampNet (and following acoustic token generation systems) now let us leverage the *meso*-scale generative properties of two-stage generative models in a live performance setting, allowing us to create and transform entire musical mesostructures like rhythms, timbral melodies and musical phrases at interactive speeds.

For a co-creative human-AI interaction, a *meso*-level generative system can be *more "generative"* than a *sound-object* level system like RAVE, while being more *interactive* than a *macro* level system like Suno, placing it a *sweet spot* between generativity (*agency)* and interactivity (*control*).

VampNet points the way to a future where large generative models are not replacements for the human creative process, but instead a medium for manipulating sonic materials – a medium powerful enough to let us interact and bend sonic material in unprecedented ways, and also an agentic medium: a medium where the materials can bite back at you and impose their own properties upon yours, if left to their own devices.

Chapter 3 introduced Sketch2Sound, a controllable audio generation method for synthesizing sounds from interpretable, time-varying control signals like loudness, pitch, and spectral centroid (i.e., *brightness*) along with text prompts. These capabilities make Sketch2Sound able to create sounds using vocal imitations as guidance for the temporal morphology of the sound. This is a novel, controllable, gestural, and expressive way of generating sounds.

Sketch2Sound opens the doors for sound designers and Foley artists to create rich sonic compositions with "human" performative gestures while still leveraging the rich and timbrally expressive

power of larger text-to-sound generation models.

I like to think of generative sound models as a "Foley-stage-in-a-box": a text-to-sound model is capable of holding hundreds of thousands of sonic materials in it. Before Sketch2Sound, text-to-sound synthesis was limited to mostly text-only conditioning. This meant samples of these sonic materials could be easily summoned via text prompts, but they could not be "played" in the same way a Foley artist plays with their props like musical instruments. Sketch2Sound paved the way so that Foley artists wanting to experiment with generative modelling systems are now able to use their voice and sonic imitations as a way to play and perform with and *beyond* the many sonic materials captured by the generative sound model.

Sketch2Sound has been announced as an upcoming feature in Adobe Firefly [1] under the name "Voice-to-sound effects". At the moment of writing, the Sketch2Sound technology has a U.S. patent pending.

Most importantly, I'd like to make the following positionality statement regarding foley sound generation research: much of the existing research in Foley sound generation [176, 177, 178] has focused on the automation of Foley sound, engendering the research task of "video-guided sound generation". This approach sidelines the human, performative nature of the craft of Foley. In contrast, my work embraces and centers the embodied, gestural process of soundmaking. Rather than replacing the sound artist, Sketch2Sound creates a new way for sound artists to manipulate the sound materials they already work with, offering them a medium through which they can engage creatively and improvisationally via sonic imitations and text guidance. I have no interest in building tools for the purpose of saving time, money, or automating an artist away from a project. Quite the opposite, Sketch2Sound is a tool for engaging in the process of soundmaking in expressive, nuanced, thorough, and most importantly *human* ways.

Chapters 4 and 5 detail my practice-based research work with two-stage generative models (specifically, with VampNet, explained in Chapter 2). Chapter 4 provided an overview of practice-based research in new musical instrument design , as well as went over some helpful background

---

[1]https://firefly.adobe.com/

material on Experimental AI music and a theoretical framework on the different perceptible time scales of music.

In Chapter 5, I introduced the neural tape loop, a co-creative generative musical meta-instrument for experimental music and sound art. I detailed four original creative works that show how the neural tape loop can be deployed in different interactive and non-interactive musicmaking formats (i.e., sound installation, live performance, fixed multichannel media), within a long-term experimental music practice. I developed new token manipulation techniques for different musical applications based on the idea of sound transformation via masking and regeneration using a masked acoustic token model (like VampNet [16]) and a sound artist's custom sound palette.

I discussed how the process of making these creative works (and engaging in collaborations with artists) led to through iterations on the design of the system and its accompanying interfaces. I reflected on how engaging in long-term musical practice with generative modeling can lead to the discovery of new techniques for playing with these models. I also discussed how sometimes technical problems may have musical solutions that can be just as satisfying and can also be employed before engaging in cognitively (and perhaps financially) expensive work aimed at solving said technical issues.

The neural tape loop contributes not just a new co-creation system and techniques to play with said system, but it is also the first practice-based research account for a sound artist working with a large, two-stage generative model, which contains uniquely new capabilities (and thus musical affordances) like the generation of musical mesostructures and the structural transfer from one sound distribution to another. The neural tape loop opens a door for more practice-based research with large generative models, and encourages generative modeling researchers to become the "first players" of the music models and co-creation systems they make. I encourage these researchers to follow their own artistic vision[2] and that of their collaborators, as musical instruments created in a hypercapitalist environment with no artistic vision are the thing that leads to the dilution and commodification of music creation.

---

[2]if the researcher feels like they have no artistic vision, I'd encourage them to find a long-term, equally contributing artistic collaborator, or, alternatively, to search for a different machine learning task altogether.

Many researchers focus on "bumping up the numbers", trying to improve the perceived "realness" of full-length generated audio tracks. In the hands of large corporations, this inevitably has led to systems that automate the production of generic *muzak* that introduces no artistic value to our communities but instead waters down[3] the existing channels we have created for engaging in musical community with one another (e.g., YouTube, SoundCloud, streaming services like Spotify and Deezer).

I've been listening to music on YouTube a lot lately. Increasingly, my feed has been contaminated with recommendations of music generated by AI artists that pump out new songs daily. Perhaps the most offensive part of this all is that the marketing and framing of these songs try to assimilate theses songs into already established musical communities – attempting to fool the listener into thinking that they're listening to (for example) reggae music made by reggae artists, when they're actually listening to AI reggae music made by an AI hustler who probably manages a dozen other "AI artists" spanning different musical traditions, each with its own revenue stream.

Instead, we should make instruments to engender new musical movements, rituals, communities and practices where generative models are one of the musical instruments of choice. Due to historic and material ties, these communities would be inextricably linked to the already existing musical practices and communities in computer music. These musical movements need to reflect and exploit the affordances of these systems (like the idea of "infinite" music processes like Dadabots' Relentless Doppelganger[4]).

I've focused on building systems that allow one to interact with the AI medium in new, more expressive ways. I share Brian Eno's belief that just like we did with vacuum tube breakup (i.e. "analog distortion") and the "crap sound of 8-bit", we'll come to love that "neural network sound" that we find ugly and weird now, as these imperfections are "the excitement of witnessing events too momentous for the medium assigned to record them".

I look forward to fostering new ways for us to engage in the beautiful rituals, practices, and cultures we call music.

---

[3]https://newsroom-deezer.com/2025/04/deezer-reveals-18-of-all-new-music-uploaded-to-stre

[4]https://www.youtube.com/watch?v=JF2p0Hlg_5U

113

REFERENCES

[1] M. Eaglin, "Ai music companies say their tools can democratize the art form. some artists are skeptical," *NBC News*, 2024.

[2] F. Iazzetta, "Meaning in musical gesture," *Trends in gestural control of music*, pp. 259–268, 2000.

[3] T. Wishart, *On Sonic Art*. Psychology Press, 1996.

[4] T. Wishart, "The composition of "vox-5"," *Computer Music Journal*, vol. 12, no. 4, pp. 21–27, 1988.

[5] G. Lemaitre and D. Rocchesso, "On the effectiveness of vocal imitations and verbal descriptions of sounds," *The Journal of the Acoustical Society of America*, vol. 135, no. 2, pp. 862–873, 2014.

[6] G. Lemaitre, O. Houix, F. Voisin, N. Misdariis, and P. Susini, "Vocal imitations of non-vocal sounds," *PloS one*, vol. 11, no. 12, e0168167, 2016.

[7] P. S. Guillaume Lemaitre Arnaud Dessein and K. Aura, "Vocal imitations and the identification of sound events," *Ecological Psychology*, vol. 23, no. 4, pp. 267–307, 2011.

[8] Y. Zhang, J. Hu, Y. Zhang, B. Pardo, and Z. Duan, "Vroom! a search engine for sounds by vocal imitation queries," in *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, 2020.

[9] D. S. Blancas and J. Janer, "Sound retrieval from voice imitation queries in collaborative databases," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, Audio Engineering Society, 2014.

[10] M Resnick, *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. Mit Press, 2017.

[11] P. Esling and N. Devis, *Creativity in the era of artificial intelligence*, 2020. arXiv: `2008.05959 [cs.CY]`.

[12] A. Caillon and P. Esling, *Rave: A variational autoencoder for fast and high-quality neural audio synthesis*, 2021. arXiv: `2111.05011 [cs.LG]`.

[13] A. McPherson, F. Morreale, and J. Harrison, "Musical instruments for novices: Comparing nime, hci and crowdfunding approaches," in *New Directions in Music and Human-*

Computer Interaction, S. Holland, T. Mudd, K. Wilkie-McKenna, A. McPherson, and M. M. Wanderley, Eds. Cham: Springer International Publishing, 2019, pp. 179–212, ISBN: 978-3-319-92069-6.

[14] M. Rodger, P. Stapleton, M. van Walstijn, M. Ortiz, and L. S. Pardue, "What makes a good musical instrument? a matter of processes, ecologies and specificities," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, R. Michon and F. Schroeder, Eds., Birmingham, UK: Birmingham City University, 2020, pp. 405–410.

[15] J. Bulley and O. Sahin, "Practice Research - Report 1: What is practice research? and Report 2: How can practice research be shared?" Practice Research Advisory Group UK (PRAG-UK), London, Report, Apr. 2021, ISBN: 9781527289079 Num Pages: 185.

[16] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, "Vampnet: Music generation via masked acoustic token modeling," in *International Society for Music Information Retrieval (ISMIR)*, 2023.

[17] H. F. Garcia, O. Nieto, J. Salamon, B. Pardo, and P. Seetharaman, "Sketch2sound: Controllable audio generation via time-varying signals and sonic imitations," in *ICASSP 2025*, 2024.

[18] M. Horta Valenzuela, *Semilla.ai*.

[19] N. Privato, V. Shepardson, G. Lepri, and T. Magnusson, "Stacco: Exploring the embodied perception of latent representations in neural synthesis," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Creative Commons Attribution 4.0 International License (CC BY 4.0), Utrecht, The Netherlands, 2024.

[20] V. Shepardson, J. Armitage, and T. Magnusson, "Notochord: A flexible probabilistic model for embodied midi performance," 2022.

[21] F. Visi, "The sophtar: A networkable feedback string instrument with embedded machine learning," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME'24)*, Luleå University of Technology, School of Music in Piteå, Snickargatan 20, 941 63 Piteå, Sweden, 2024.

[22] P. Bauman, *Mat Dryhurst on Becoming Infinite — lerandom.art*, `https://www.lerandom.art/editorial/mat-dryhurst-on-becoming-infinite`, [Accessed 29-10-2024], 2025.

[23] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, and C. J. Cai, "Ai song contest: Human-ai co-creation in songwriting," *arXiv preprint arXiv:2010.05388*, 2020.

[24] R. Louie, A. Coenen, C. Z. Huang, M. Terry, and C. J. Cai, "Novice-ai music co-creation via ai-steering tools for deep generative models," in *Proceedings of the 2020 CHI Confer-*

ence on Human Factors in Computing Systems, ser. CHI '20, ¡conf-loc¿, ¡city¿Honolulu¡/city¿, ¡state¿HI¡/state¿, ¡country¿USA¡/country¿, ¡/conf-loc¿: Association for Computing Machinery, 2020, 1–13, ISBN: 9781450367080.

[25] R. Louie, J. Engel, and C.-Z. A. Huang, "Expressive communication: Evaluating developments in generative models and steering interfaces for music creation," in *27th International Conference on Intelligent User Interfaces*, ser. IUI '22, Helsinki, Finland: Association for Computing Machinery, 2022, 405–417, ISBN: 9781450391443.

[26] S. J. Krol, M. T. L. Rodriguez, and M. L. Paredes, *Exploring the Needs of Practising Musicians in Co-Creative AI Through Co-Design*, arXiv:2502.09055 [cs], Feb. 2025.

[27] T. H. Park, "Instrument technology: Bones, tones, phones, and beyond," in *The Routledge Companion to Music, Technology, and Education*, Routledge, 2017, pp. 39–46.

[28] B. L. T. Sturm *et al.*, "Ai Music Studies: Preparing for the Coming Flood," *AIMC 2024 (09/09 - 11/09)*, 2024, https://aimc2024.pubpub.org/pub/ej9b5mv1.

[29] F. Morreale, S. M. A. Bin, A. McPherson, P. Stapleton, and M. Wanderley, "A nime of the times: Developing an outward-looking political agenda for this community," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, R. Michon and F. Schroeder, Eds., Birmingham, UK: Birmingham City University, 2020, pp. 160–165.

[30] J. Roberts, "Art after deskilling," *Historical Materialism*, vol. 18, no. 2, pp. 77–96, 2010.

[31] B. Krause, "The niche hypothesis: A virtual symphony of animal sounds, the origins of musical expression and the health of habitats," *Soundscape Newsletter (World Forum for Acoustic Ecology)*, Jun. 1993.

[32] P. Alperson, "The instrumentality of music," *Journal of Aesthetics and Art Criticism*, vol. 66, no. 1, pp. 37–51, 2008.

[33] S. Hardjowirogo, "Instrumentality. on the construction of instrumental identity," in Dec. 2017, pp. 9–24, ISBN: 978-981-10-2950-9.

[34] D. Cavdir, "Touch, Listen, (Re)Act: Co-designing Vibrotactile Wearable Instruments for Deaf and Hard of Hearing," in *NIME 2022*, https://nime.pubpub.org/pub/uyoq7iv0, 2022.

[35] J. Sullivan, J. Vanasse, C. Guastavino, and M. Wanderley, "Reinventing the noisebox: Designing embedded instruments for active musicians," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, R. Michon and F. Schroeder, Eds., Birmingham, UK: Birmingham City University, 2020, pp. 5–10.

[36] J. Armitage, T. Magnusson, V. Shepardson, and H. Ulfarsson, "The Proto-Langspil: Launching an Icelandic NIME Research Lab with the Help of a Marginalised Instrument," in *NIME 2022*, https://nime.pubpub.org/pub/langspil, 2022.

[37] SupaduDev, *Forty Years of Computer Music Journal*, Jun. 2016.

[38] S. Fasciani and J. Goode, "20 nimes: Twenty years of new interfaces for musical expression," in *NIME 2021*, PubPub, 2021.

[39] M. A. J. Baalman, "Interplay between composition, instrument design and performance," in *Musical Instruments in the 21st Century: Identities, Configurations, Practices*, T. Bovermann, A. de Campo, H. Egermann, S.-I. Hardjowirogo, and S. Weinzierl, Eds. Singapore: Springer Singapore, 2017, pp. 225–241, ISBN: 978-981-10-2951-6.

[40] C. Small, *Musicking: The meanings of performing and listening*. Wesleyan University Press, 1998.

[41] L. Bittencourt, "Reflections on live looping and instrumentality through the performance of import/export: Percussion suite for global junk by gabriel prokofiev," *Live Looping in Musical Performance: Lusophone Experiences in Dialogue*, 2023.

[42] A. F. Blackwell and N. M. Collins, "The programming language as a musical instrument," in *Annual Workshop of the Psychology of Programming Interest Group*, 2005.

[43] P. A. Nilsson, "A field of possibilities: Designing and playing digital musical instruments," Ph.D. dissertation, Göteborgs Universitet, 2011.

[44] J. Bowers and O. Green, "All the Noises: Hijacking Listening Machines for Performative Research," *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2018.

[45] T. Pelinski, A. McPherson, and R. Fiebrink, "Ways of knowing, ways of writing: Technical practice research in new musical instrument design," *Journal of New Music Research*, pp. 1–14, Jan. 2025.

[46] J. C. Schacher, "Gestural performance of electronic music—a "nime" practice as research," *Leonardo*, vol. 49, no. 1, pp. 84–85, 2016.

[47] O. Green, "NIME, Musicality and Practice-led Methods," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2014.

[48] M. Gurevich, "Diversity in nime research practices," *Leonardo*, vol. 49, no. 1, pp. 80–81, 2016.

[49] N. Howell, A. Desjardins, and S. Fox, "Cracks in the success narrative: Rethinking failure in design research through a retrospective trioethnography," *ACM Trans. Comput.-Hum. Interact.*, vol. 28, no. 6, Nov. 2021.

[50] S. Dieleman. "Generative modeling in the latent space." (2025), (visited on 04/21/2025).

[51] A. Ramesh *et al.*, "Zero-shot text-to-image generation," in *International conference on machine learning*, Pmlr, 2021, pp. 8821–8831.

[52] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[53] H. Liu *et al.*, "AudioLDM: Text-to-audio generation with latent diffusion models," *Proceedings of the International Conference on Machine Learning*, 2023.

[54] Z. Borsos *et al.*, *Audiolm: A language modeling approach to audio generation*, 2023. arXiv: 2209.03143 [cs.SD].

[55] A. Agostinelli *et al.*, *Musiclm: Generating music from text*, 2023. arXiv: 2301.11325 [cs.SD].

[56] J. Copet *et al.*, "Simple and controllable music generation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[57] A. Ziv *et al.*, *Masked audio generation using a single non-autoregressive transformer*, 2024. arXiv: 2401.04577 [cs.SD].

[58] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, *High-fidelity audio compression with improved rvqgan*, 2023. arXiv: 2306.06546 [cs.SD].

[59] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. arXiv: http://arxiv.org/abs/1312.6114v10 [stat.ML].

[60] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, 495–507, 2021.

[61] H. Liu *et al.*, "AudioLDM 2: Learning holistic audio generation with self-supervised pre-training," *arXiv preprint arXiv:2308.05734*, 2023.

[62] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, "Fast timing-conditioned latent audio diffusion," in *International Conference on Machine Learning (ICML)*, 2024.

[63] D. Yang *et al.*, "Diffsound: Discrete diffusion model for text-to-sound generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1720–1733, 2023.

[64] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, *Soundstorm: Efficient parallel audio generation*, 2023. arXiv: `2305.09636 [cs.SD]`.

[65] Z. Evans, J. D. Parker, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, *Long-form music generation with latent diffusion*, 2024. arXiv: `2404.10301 [cs.SD]`.

[66] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 315–11 325.

[67] M. Bavarian *et al.*, "Efficient training of language models to fill in the middle," *arXiv preprint arXiv:2207.14255*, 2022.

[68] Z. Novack, G. Zhu, J. Casebeer, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, *Presto! distilling steps and layers for accelerating music generation*, 2024. arXiv: `2410.05167 [cs.SD]`.

[69] M. Pasini, J. Nistal, S. Lattner, and G. Fazekas, "Continuous autoregressive models with noise augmentation avoid error accumulation," *arXiv preprint arXiv:2411.18447*, 2024.

[70] J. D. Parker *et al.*, "Stemgen: A music generation model that listens," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2024.

[71] Y. Chung, J. Lee, and J. Nam, *T-foley: A controllable waveform-domain diffusion model for temporal-event-guided foley sound synthesis*, 2024. arXiv: `2401.09294 [cs.SD]`.

[72] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, "DITTO: Diffusion inference-time t-optimization for music generation," in *International Conference on Machine Learning (ICML)*, 2024.

[73] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 32, 2692–2703, 2024.

[74] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg, "Structured denoising diffusion models in discrete state-spaces," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 981–17 993, 2021.

[75] Y. Yuan, H. Liu, X. Liu, Q. Huang, M. D. Plumbley, and W. Wang, "Retrieval-augmented text-to-audio generation," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 581–585.

[76] R. Huang *et al.*, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," in *International Conference on Machine Learning*, PMLR, 2023, pp. 13 916–13 932.

[77] F. Kreuk *et al.*, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.

[78] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction guided latent diffusion model," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23, Ottawa ON, Canada: Association for Computing Machinery, 2023, 3590–3598, ISBN: 9798400701085.

[79] J. Huang *et al.*, *Make-an-audio 2: Temporal-enhanced text-to-audio generation*, 2023. arXiv: `2305.18474 [cs.SD]`.

[80] H. Liu *et al.*, "Audiolcm: Text-to-audio generation with latent consistency models," *arXiv preprint arXiv:2406.00356*, 2024.

[81] J. Xue, Y. Deng, Y. Gao, and Y. Li, "Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 4700–4712, 2024.

[82] H.-H. Wu, O. Nieto, J. P. Bello, and J. Salamon, "Audio-text models do not yet leverage natural language," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[83] D. Smalley, "Spectromorphology: Explaining sound-shapes," *Organised sound*, vol. 2, no. 2, pp. 107–126, 1997.

[84] O. Tal, A. Ziv, I. Gat, F. Kreuk, and Y. Adi, *Joint audio and symbolic conditioning for temporally controlled text-to-music generation*, 2024. arXiv: `2406.10970 [cs.SD]`.

[85] M. Morrison, C. Churchwell, N. Pruyne, and B. Pardo, "Fine-grained and interpretable neural speech editing," in *Interspeech 2024*, 2024.

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

[87] D.-Y. Wu *et al.*, "Ddsp-based singing vocoders: A new subtractive-based synthesizer and a comprehensive evaluation," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India: Creative Commons Attribution 4.0 International License (CC BY 4.0), 2022.

[88] A. Neely, *Turning bass into violin (using ai)*, YouTube video, 1.78M subscribers, 304,233 views, Published on Oct 20, 2020, Accessed on Oct 22, 2024, 2020.

[89] F. Caspe, A. McPherson, and M. Sandler, "Ddx7: Differentiable fm synthesis of musical instrument sounds," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Attribution: F. Caspe, A. McPherson, and M. Sandler, "DDX7: Differentiable FM Synthesis of Musical Instrument Sounds.", Bengaluru, India: Creative Commons Attribution 4.0 International License (CC BY 4.0), 2022.

[90] V. Shepardson and T. Magnusson, "The living looper: Rethinking the musical loop as a machine action-perception loop," M. Ortiz and A. Marquez-Borbon, Eds., pp. 224–231, 2023.

[91] T. Pelinski, R. Diaz, A. L. B. Temprano, and A. McPherson, "Pipeline for recording datasets and running neural networks on the bela embedded hardware platform," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Mexico City, Mexico: Creative Commons Attribution 4.0 International License (CC BY 4.0), 2023.

[92] N. Devis, N. Demerlé, S. Nabi, D. Genova, and P. Esling, "Continuous descriptor-based control for deep audio synthesis," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[93] H. Davies, "A history of sampling," *Org. Sound*, vol. 1, no. 1, 3–11, Apr. 1996.

[94] P. A. Tremblay *et al.*, *Fluid corpus manipulation toolbox (v.1)*, 2022.

[95] G. Roma, P. A. Tremblay, and O. Green, "Graph-based audio looping and granulation," in *2021 24th International Conference on Digital Audio Effects (DAFx)*, 2021, pp. 253–259.

[96] G. Roma, A. Xambó, O. Green, and P. A. Tremblay, "A general framework for visualization of sound collections in musical interfaces," *Applied Sciences*, vol. 11, no. 24, 2021.

[97] J. Bell, "Maps as scores: "timbre space" representations in corpus-based concatenative synthesis," in *International Conference on Technologies for Music Notation and Representation (TENOR)*, ⟨hal-04182706v2⟩, Northeastern University, Boston, United States, 2023.

[98] P. E. Benjamin Hackbarth Norbert Schnell and D. Schwarz, "Composing morphology: Concatenative synthesis as an intuitive medium for prescribing sound in time," *Contemporary Music Review*, vol. 32, no. 1, pp. 49–59, 2013. eprint: https://doi.org/10.1080/07494467.2013.774513.

[99] D. Schwarz, "The sound space as musical instrument: Playing corpus-based concatenative synthesis," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan: University of Michigan, 2012.

[100] L. Garber, T. Ciccola, and J. C. Amusategui, "Audiostellar, an open source corpus-based musical instrument for latent sound structure discovery and sonic experimentation," Dec. 2020.

[101] D. Schwarz, "Corpus-based concatenative synthesis," *IEEE signal processing magazine*, vol. 24, no. 2, pp. 92–104, 2007.

[102] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[103] C. Wang *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[104] Z. Borsos *et al.*, "Audiolm: A language modeling approach to audio generation," *arXiv preprint arXiv:2209.03143*, 2022.

[105] D. Rampas, P. Pernias, E. Zhong, and M. Aubreville, "Fast text-conditional discrete denoising on vector-quantized latent spaces," *arXiv preprint arXiv:2211.07292*, 2022.

[106] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[107] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*, 2022.

[108] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[109] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[110] C. Gârbacea *et al.*, "Low bit-rate speech coding with vq-vae and a wavenet decoder," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 735–739.

[111] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[112] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[113] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

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

[115] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[116] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[117] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[118] Y.-A. Chung *et al.*, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2021, pp. 244–250.

[119] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[120] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[121] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning," *Advances in neural information processing systems*, vol. 30, 2017.

[122] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[123] H. Chang *et al.*, "Muse: Text-to-image generation via masked generative transformers," *arXiv preprint arXiv:2301.00704*, 2023.

[124] E. J. Gumbel, *Statistical theory of extreme values and some practical applications; a series of lectures.* Washington, 1954.

[125] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[126] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017. arXiv: 1711.05101.

[127] C. J. Steinmetz and J. D. Reiss, "WaveBeat: End-to-end beat and downbeat tracking in the time domain," in *151st AES Convention*, 2021.

[128] J. Barnett, "The ethical implications of generative audio models: A systematic literature review," in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '23, Montréal, QC, Canada: Association for Computing Machinery, 2023, 146–161, ISBN: 9798400702310.

[129] E. Frid, C. Gomes, and Z. Jin, "Music creation by example," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ACM, 2020, pp. 1–13.

[130] J. Barnett, H. F. Garcia, and B. Pardo, "Exploring musical roots: Applying audio embeddings to empower influence attribution for a generative music model," *arXiv preprint arXiv:2401.14542*, 2024.

[131] M. Comunità *et al.*, "Specmaskgit: Masked generative modeling of audio spectrograms for efficient audio synthesis and beyond," *arXiv preprint arXiv:2406.17672*, 2024.

[132] P. O'Reilly, H. F. Garcia, P. Seetharaman, and B. Pardo, "Masked token modeling for zero-shot anything-to-drums conversion," in *Extended Abstracts for the Late-Breaking Demo Session of the 25th Int. Society for Music Information Retrieval Conf.*, 2024.

[133] V. Ament, *The Foley Grail: The Art of Performing Sound for Film, Games, and Animation*, 3rd. Routledge, 2021.

[134] Z. Xie, X. Xu, Z. Wu, and M. Wu, "Picoaudio: Enabling precise timestamp and frequency controllability of audio events in text-to-audio generation," *arXiv preprint arXiv:2407.02869*, 2024.

[135] S. Delle Monache, D. Rocchesso, F. Bevilacqua, G. Lemaitre, S. Baldan, and A. Cera, "Embodied sound design," *International Journal of Human-Computer Studies*, vol. 118, pp. 47–59, 2018.

[136] A. Hazan, "Billaboop: Real-time voice-driven drum generator," in *Audio Engineering Society Convention 118*, Audio Engineering Society, 2005.

[137] J. Janer and M. De Boer, "Extending voice-driven synthesis to audio mosaicing," in *5th Sound and Music Computing Conference, Berlin*, Citeseer, vol. 4, 2008.

[138] A. D. Piccolo and D. Rocchesso, "Non-speech voice for sonic interaction: A catalogue," *Journal on Multimodal User Interfaces*, vol. 11, pp. 39–55, 2017.

[139] S. Fasciani and L. Wyse, "Mapping the voice for musical control," 2013.

[140] S. Baldan, S. Delle Monache, D. Rocchesso, H. Lachambre, *et al.*, "Sketching sonic interactions by imitation-driven sound synthesis," in *Proceedings of the 13. Sound & Music Computing Conference*, Zentrum für Mikrotonale Musik und Multimediale Komposition (ZM4) Hochschule ..., 2016, pp. 47–54.

[141] S. Zhao *et al.*, "Uni-controlnet: All-in-one control to text-to-image diffusion models," *Advances in Neural Information Processing Systems*, 2023.

[142] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[143] M. Morrison, *Torchcrepe*, 2022.

[144] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *CVPR*, 2023.

[145] B. Kim, M. Cartwright, and B. Pardo, *Vimsketch dataset*, 2019.

[146] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fr\'echet audio distance: A metric for evaluating music enhancement algorithms," *Interspeech*, 2018.

[147] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, "Adapting frechet audio distance for generative music evaluation," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1331–1335.

[148] S. Hershey *et al.*, "Cnn architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[149] Y. Wu*, K. Chen*, T. Zhang*, Y. Hui*, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[150] C. Roads, *Microsound*. The MIT Press, 2004, ISBN: 0262681544.

[151] Lemelson Center for the Study of Invention and Innovation, *Invention: Electric guitar*, `https://web.archive.org/web/20180824234551/http://invention.si.edu/invention-electric-guitar/p/35-invention`, Originally published on invention.si.edu. Archived on August 24, 2018, 2018.

[152] D. Blackeye, *Distortion – the physics of heavy metal*, `https://web.archive.org/web/201001060700113/https://www.bbc.co.uk/dna/h2g2/A33659319`, Originally published on BBC h2g2, May 1, 2008. Accessed via the Wayback Machine on January 6, 2010, 2008.

[153] M. Campbell, C. Greated, and A. Myers, *Musical Instruments: History, Technology, and Performance of Instruments of Western Music*. Oxford: Oxford University Press, 2004, Online edn, Oxford Academic, 1 May 2008. Accessed 3 May 2025.

[154] P. Cook, "Principles for designing computer music controllers," in *Proceedings of the 2001 Conference on New Interfaces for Musical Expression*, ser. NIME '01, Seattle, Washington: National University of Singapore, 2001, 1–4.

[155] S. J. and, "Instruments and players: Some thoughts on digital lutherie," *Journal of New Music Research*, vol. 33, no. 3, pp. 321–341, 2004.

[156] T. Pelinski *et al.*, "Embedded AI for NIME: Challenges and Opportunities," *International Conference on New Interfaces for Musical Expression*, 2022, https://nime.pubpub.org/pub/rwr2c3zs.

[157] G. E. Lewis, "Too many notes: Computers, complexity and culture in "voyager"," *Leonardo Music Journal*, vol. 10, pp. 33–39, 2000.

[158] S. O'modhrain, "A framework for the evaluation of digital musical instruments," *Computer Music Journal*, vol. 35, no. 1, pp. 28–42, 2011.

[159] B. Carey and A. Johnston, "Reflection on action in nime research: Two complementary perspectives," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Brisbane, Australia: Queensland Conservatorium Griffith University, 2016, pp. 377–382, ISBN: 978-1-925455-13-7.

[160] L. Elblaus, K. Hansen, and R. Bresin, "Nime design and contemporary music practice: Benefits and challenges," in *Practice-Based Research Workshop held at the International Conference on New Interfaces for Musical Expression*, vol. 30, 2014.

[161] C. Neustaedter and P. Sengers, "Autobiographical design in hci research: Designing and learning through use-it-yourself," in *Proceedings of the Designing Interactive Systems Conference*, ser. DIS '12, Newcastle Upon Tyne, United Kingdom: Association for Computing Machinery, 2012, 514–523, ISBN: 9781450312103.

[162] C. Robson and K. McCartan, *Real world research*. Blackwell Oxford, 2002, vol. 2.

[163] C. Gray, "Inquiry through practice: Developing appropriate research strategies," in *International Conference on Art and Design Research: No Guru*, 1998.

[164] S. Scrivener, "Reflection in and on action and practice in creative-production doctoral projects in art and design," *Working Papers in art and design*, vol. 1, no. 1, n–pag, 2000.

[165] B. Carey, "$_derivations and the performer-developer$: Co-evolving digital artefacts and human-machine performance practices," Ph.D. dissertation, University of Technology, Sydney, Feb. 2016.

[166] L. Dahl, "Designing new musical interfaces as research: What's the problem?" *Leonardo*, vol. 49, no. 1, pp. 76–77, 2015.

[167] Y. Rubinstein, "Uneasy listening: Towards a hauntology of ai-generated music," *Resonance: The Journal of Sound and Culture*, vol. 1, no. 1, pp. 77–93, 2020.

[168] N. Privato and T. Magnusson, "Querying the ghost: Ai hauntography in nime," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, NIME, Utrecht, The Netherlands: Intelligent Instruments Lab, University of Iceland, 2024.

[169] R. Dudas, ""Comprovisation": The Various Facets of Composed Improvisation within Interactive Performance Systems," *Leonardo Music Journal*, vol. 20, pp. 29–31, Dec. 2010, _eprint: https://direct.mit.edu/lmj/article-pdf/doi/10.1162/LMJ_a_00009/1675112/lmj_a_00009.pdf.

[170] R. Fiebrink, D. Trueman, and P. R. Cook, "A meta-instrument for interactive, on-the-fly machine learning," in *New Interfaces for Musical Expression*, 2009.

[171] G. Vigliensoni, P. Perry, R. Fiebrink, *et al.*, "A small-data mindset for generative ai creative work," 2022.

[172] B. McFee *et al.*, "Librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, K. Huff and J. Bergstra, Eds., 2015, pp. 18–25.

[173] C. Meister, T. Pimentel, G. Wiher, and R. Cotterell, "Locally typical sampling," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 102–121, 2023.

[174] C. Malloy, "Timbral effects the paulstretch audio time-stretching algorithm," *The Journal of the Acoustical Society of America*, vol. 151, no. 4_Supplement, A158–A158, 2022.

[175] M. Dwyer, "This musician was getting millions of streams. then fake tracks appeared under his name," *The Sydney Morning Herald*, May 28, 2025, Accessed: 2025-05-29.

[176] Z. Zhong, A. Takahashi, S. Cui, K. Toyama, S. Takahashi, and Y. Mitsufuji, "Specmask-foley: Steering pretrained spectral masked generative transformer toward synchronized video-to-audio synthesis via controlnet," *arXiv preprint arXiv:2505.16195*, 2025.

[177] J. Lee, J. Im, D. Kim, and J. Nam, "Video-foley: Two-stage video-to-sound generation via temporal event condition for foley sound," *arXiv preprint arXiv:2408.11915*, 2024.

[178] S. Luo, C. Yan, C. Hu, and H. Zhao, "Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 48 855–48 876, 2023.