Designing novel time-frequency scales for interactive music creation with hierarchical statistical modeling

Théis Bazin

DESIGNING NOVEL TIME-FREQUENCY SCALES FOR INTERACTIVE MUSIC CREATION WITH HIERARCHICAL STATISTICAL MODELING

THÉIS BAZIN

SONY CSL PARIS

IRCAM, UMR STMS 9912, SORBONNE UNIVERSITÉ

AN ARTIST-CENTRIC APPROACH TO AI-ASSISTED MUSIC CREATION PROTOTYPE DESIGN

In Fulfillment of the Requirements for the degree of
Doctor of Philosophy in Computer Science

Defended on May 31st, 2023 at IRCAM

Industrial Supervisor	
Dr. Gaëtan HADJERES	Senior Research Scientist, SonyAI
Academic Supervisor	
Dr. Mikhail MALT	Researcher – RepMus, STMS, IRCAM, Sorbonne Université, CNRS
Co-supervisor	
Prof. Philippe Esling	Associate Professor – IRCAM
President of the jury	
Dr. Jean BRESSON	Research Director, RepMus, STMS, IRCAM, Sorbonne Université, CNRS – Team Product Owner, Ableton
Reviewers	
Prof. Wendy MACKAY	Research Director, Classe Exceptionnelle, INRIA-Saclay, ex-situ research group
Prof. Geoffroy PEETERS	Full Professor, Image-Data-Signal (IDS) department, LTCI, Telecom Paris, IP Paris
Examiner	
Prof. Cheng-Zhi Anna HUANG	Adjunct Professor, Université de Montréal – Research Scientist, Google – Canada CIFAR AI Chair

Théis Bazin: *Designing novel time-frequency scales for interactive music creation with hierarchical statistical modeling*, An artist-centric approach to AI-assisted music creation prototype design, © October 3, 2023

To Môm in.
I would likely not be here if not for you.

To Dr. Squirrel and Mimitutu, for your guidance and friendship.
I promise you, one day I will train that new version of NOTONO.

Above all, this is dedicated to Quitterie, for your support and love
and for sticking with me through the hard times.
You deserve a fair share of the credits for this,
This is your PhD as much as it is mine.

DECLARATION

This work was supported by the French Association Nationale de la Recherche et de la Technologie through the CIFRE PhD contract number 2019.0094.

ABSTRACT

Modern musical creation unfolds on many different time scales: from the vibration of a string or the resonance of an electronic instrument at the millisecond scale, through the few seconds typical of an instrument's note, to the tens of minutes of operas or DJ sets. The interleaving of these multiple scales has led to the development of numerous technical and theoretical tools to ease the manipulation of time. These abstractions, such as scales, rhythmic notations, or even usual models of audio synthesis, largely infuse current tools – software and hardware – for musical creation. However, these abstractions, which emerged for the most part during the 20th century in the West on the basis of classical musical theories of written music, are not devoid of cultural assumptions. They reflect various principles aimed at abstracting away certain aspects of the music, whose high degree of physical variability makes them typically inconvenient for musical writing. Examples of these include micro-deviations with respect to a metronomic time or micro-deviations of frequency with respect to an idealized pitch.

These compromises, typically relevant when the written music is intended to be performed by musicians, able to reintroduce subtle variations and musical richness, are however limiting in the context of computer-assisted music creation, wherein computers coldly render these coarse abstractions, ultimately tending to restrict the diversity of the music that can be produced with these tools. Through a review of several typical interfaces for music creation, I show that an essential factor is the *scale* of the human-machine interactions proposed by these abstractions. At their most flexible level, such as audio representations or piano-roll representations with unquantized time, they prove difficult to manipulate, as they require a high degree of precision, particularly unsuitable for modern mobile and touch terminals. On the other hand, in most commonly used abstractions with discretized time, such as scores or sequencers, they prove to be too constraining for the creation of culturally diverse music that does not follow the proposed time and pitch grids.

In this thesis, I argue that artificial intelligence, through its ability to build high-level representations of given complex objects, allows

the construction of new scales of music creation, designed *for* interaction, and thus enables radically new approaches to music creation. I present and illustrate this idea through the design and implementation of three web-based prototypes of music creation assisted by artificial intelligence, one of which is based on a new neural model for the inpainting of musical instrument sounds, a model which was I designed, implemented and trained in the context of this thesis. These high-level representations – for sheet music, piano-rolls, and spectrograms – are deployed at a time-frequency scale coarser than the original data, but better suited to interaction. By allowing localized transformations on these representations but also capturing, through statistical modeling, aesthetic specificities and fine micro-variations of the original musical training data, these tools allow to easily and controllably obtain musically rich results. Through the evaluation of these three prototypes in real conditions by several artists, I show that these new scales of interactive creation are useful for both experts and novices. Thanks to the assistance of AI on technical aspects that normally require precision and expertise, they are also suitable for use on touch screens and mobile devices.

RÉSUMÉ

La création musicale moderne se déploie à de nombreuses échelles de temps différentes : de la vibration d'une corde ou la résonance d'un instrument électronique à l'échelle de la milliseconde en passant par les quelques secondes typiques d'une note d'instrument, jusqu'aux dizaines de minutes d'opéras ou de DJ sets. L'entremêlement de ces multiples échelles a mené au développement de nombreux outils techniques et théoriques pour rendre efficace cette entreprise de manipulation du temps. Ces abstractions, telles les gammes, les notations rythmiques ou encore les modèles courants de synthèse audio, influencent fortement les outils actuels – logiciels et matériels – de création musicale. Pourtant, ces abstractions, qui ont émergé pour la plupart au cours du 20ème siècle en Occident sur la base de théories musicales classiques de la musique écrite, ne sont pas dénuées d'a priori culturels. Elles reflètent des principes déterminés visant à gommer certains aspects de la musique (par exemple, micro-déviations par rapport à un temps métronomique ou micro-déviations de fréquence par rapport à une hauteur idéalisée), dont le haut degré de variabilité physique les rend typiquement peu commodes pour l'écriture musicale.

Ces compromis, qui s'avèrent pertinents lorsque la musique écrite est destinée à l'interprétation par des musicien-ne-s, à même de réintroduire variations et richesse physique et musicale, se révèlent cependant limitants dans le cadre de la création musicale assistée par ordinateur, restituant froidement ces abstractions et tendent à restreindre la diversité des musiques qu'il est possible de produire avec l'outil informatique. À travers la présentation de plusieurs interfaces typiques de la création musicale, je montre qu'un facteur essentiel est l'*échelle* des interactions humain-machine proposées par ces abstractions. Pour les plus flexibles d'entre elles, telles les représentations audio ou les *piano-rolls* sur un temps non quantifié, elles se révèlent difficiles à manipuler, car elles requièrent un haut degré de précision, particulièrement inadapté aux terminaux mobiles et tactiles modernes. A contrario, dans de nombreuses abstractions à temps discrétisé communément employées comme les partitions ou les séquenceurs, elles

se révèlent contraignantes pour la création de musiques de cultures non-occidentales.

Dans cette thèse, je soutiens que l'intelligence artificielle, par la capacité qu'elle offre à construire des représentations haut-niveau d'objets complexes, permet de construire de nouvelles échelles de la création musicale, pensées *pour* l'interaction, et de proposer ainsi des approches radicalement neuves de cette création. Je présente et illustre cette idée à travers la conception et le développement de trois prototypes web de création musicales assistés par IA, dont un basé sur un modèle neuronal nouveau pour l'*inpainting* de sons d'instruments de musique, modèle également conçu et implémenté dans le cadre de cette thèse. Ces représentations haut-niveau – pour les partitions, les piano-rolls et les spectrogrammes – se déploient à une échelle temps-fréquence plus grossière que les données d'origine, mais mieux adaptée à l'interaction. En permettant d'effectuer des transformations localisées sur cette représentation mais en capturant également, par la modélisation statistique, des spécificités esthétiques et micro-variations des données musicales d'entraînement, ces outils permettent d'obtenir aisément et de façon contrôlable des résultats musicalement riches. À travers l'évaluation en conditions réelles par plusieur-e-s artistes de ces trois prototypes, je montre que ces nouvelles échelles de création interactive sont utiles autant pour les expert-e-s que pour les novices. Grâce à l'assistance de l'IA sur des aspects techniques nécessitant normalement précision et expertise, elles se prêtent de plus à une utilisation sur écrans tactiles et mobiles.

PUBLICATIONS

[1] **Bazin, T.**, Hadjeres, G., "NONOTO: A Model-agnostic Web Interface for Interactive Music Composition by Inpainting." In: *Proceedings of the Tenth International Conference on Computational Creativity, ICCC 2019, Charlotte, North Carolina, USA, June 17-21, 2019*. Ed. by Kazjon Grace, Michael Cook, Dan Ventura, and Mary Lou Maher. Association for Computational Creativity (ACC), 2019, pp. 89–91. URL: http://computationalcreativity.net/iccc2019/papers/iccc19-paper-27.pdf (visited on 12/03/2021).

[2] **Bazin, T.**, Hadjeres, G., Esling, P., Malt, M., "Spectrogram Inpainting for Interactive Generation of Instrument Sounds." In: *Proceedings of the 2020 Joint Conference on AI Music Creativity*. Stockholm, Norway: KTH Royal Institute of Technology, July 2020. DOI: 10.30746/978-91-519-5560-5. URL: https://hal.archives-ouvertes.fr/hal-03207968 (visited on 12/03/2021).

[3] **Bazin, T.**, Hadjeres, G., Malt, M., "AI-driven, Mobile-First Web UI for Controllable Expressive Piano Performance Composition." In: *Extended Abstracts for the Late-Breaking Demo Session of the 23rd Int. Society for Music Information Retrieval Conf*. ISMIR Late-Breaking Demo Session. Bengaluru, India, 2022.

[4] Esling, P., **Bazin, T.**, Bitton, A., Carsault, T., Devis, N., "Ultra-Light Deep MIR by Trimming Lottery Tickets." In: International Symposium on Music Information Retrieval (ISMIR). Oct. 15, 2020. URL: https://hal.archives-ouvertes.fr/hal-03208026 (visited on 12/06/2021).

Just as water, gas, and electricity are brought into our houses from far off to satisfy our needs in response to a minimal effort, so we shall be supplied with visual or auditory images, which will appear and disappear at a simple movement of the hand, hardly more than a sign.

Paul Valery as quoted in Walter Benjamin's *The Work of Art in the Age of Mechanical Reproduction*

I see a red door and I want to paint it black.

The Rolling Stones, in*Paint It Black*[a]

a Sir Mick Jagger is likely telling us here he wishes he had access to more inpainting-based music creations assistants. Well, despair no more, Mick!

Il jouait du PIANOTO debout
C'est peut-être un détail pour vous

Hommage à Léopold Crestel
DR. SQUIRREL

ACKNOWLEDGMENTS

Heartfelt thanks to Gaëtan for everything working with you has taught me and for generally being an outstanding tutor and a mesmerizing person. Michael "Mimitutu" Turbot, you have been a fabulous mentor and I feel lucky to know you.

Many thanks to Mikhail for accepting to direct this PhD and make it a reality, for all the pleasant discussions on Computer Music and for all your kind support along the way.

I thank Prof. Philippe Esling for helping me initiate this PhD through his help on the administrative side of it.

I write these words after the defense of the thesis, which took place at IRCAM on May 31st, 2023. I am deeply grateful to all members of my jury for gifting me their time and attention, and for their insightful remarks on the work presented here.

PHDS ARE HARD To potential prospective PhD students reading this work (if that is you, then bless you, I did certainly not expect many people to read this!), a few words of encouragement are to be found in the next paragraphs.

Regardless of the quality or lack thereof of the work presented here, it is true that we mostly ever get to witness research projects in a fully finished fashion, and only rarely get a glimpse of the hardships their authors have gone through, and this manuscript might be perceived as such by someone: the result of a pleasant and uneventful journey. This once prompted a wise friend of mine to observe that really *any* sort of *finished* work from other people can seem overwhelming from one's own perspective. It certainly has been the case for me, even more so with the tendency in academic-related social network circles to only present successes and achievements and omit hardships and failures. To put it simply, I have time and time again felt out of touch with the global course of research, and looking at others present their work and career-paths on the web making it appear like it's just success-after-success, has left me wondering about my own value and whether I'd ever manage to achieve anything remotely comparable to those – or even, well, anything at all. I therefore take this occasion to share some of the less-good/bad aspects of my own work.

Going through this PhD has been *hard*. On top of having to deal with the general situation regarding COVID, I had the misfortune of being faced with surprinsigly toxic behaviors inside my work environment. Sadly, this appears to be far from being an isolated situation, as revealed, studied and discusses in a 2022 book by French researcher Dr. Adèle B. Combes titled *Comment l'université broie les jeunes chercheurs* [22]. Ensuingly, I went through a long period of depression or near-depression which could well have resulted in me prematurely quiting the PhD.

SEEK THE SUPPORT OF THE PEOPLE WHO WILL EMPOWER YOU If you ever find yourself in such a situation, I encourage you with all my heart to move away from the people who try to bring you down and instead seek the support of those who are ready to give it to you – supportive colleagues, coworkers and senior roles, friends, relatives...

Research is demanding and it may seem like a highly individualistic and competitive endeavor. Yet, it can also be a fantastic environment for meeting inspiring, like-minded, passion-driven people. And there are actually *many* people in academia, maybe not in your own institution, but in other labs: find the ones who inspire you, and contact them via mail or their public social network profiles to ask for some guidance or for organizing a short discussion. You may be positively surprised at how many may answer and offer a moment of their time. Do not waste your energy and enthusiasm trying to please or impress the ones who will attempt to undermine you.

In this perspective, I am grateful to Prof. Rebecca Fiebrink and Prof. Anna Huang, whose respective projects have been major sources of inspiration and enjoyment for me, for the occasions they have given me to discuss ideas, projects and for all the support, invaluable hints and encouragement they have provided me on those occasions.

I have also had the privilege of working with many talented musicians as part of the multiple collaborations organized at Sony CSL Paris. Uèle (Uèle Lamore), Jérémy (Whim Therapy), Simon (CHATON), Cécile (DeLaurentis), Guillaume (Kadebostany), working with you has been one of the key motivations for pushing through and driving these tools forward. Thank you all very much for your dedication, your enthusiasm, your patience and your friendship. I feel very lucky to have had the opportunity of working with all of you and these have been invaluable moments.

I equally thank Aline (Gorisse) and Youssra (Khechai) from Y/A for expressing interest in NOTONO and volunteering to try it out in a real context. It was heart-warming to get to collaborate in such a natural and spontaneous way with friends like you, and your feedback was precious. May your respective and associated musical careers flourish!

SOME FANTASTIC TEACHERS AND INSTITUTIONS Over the course of my curriculum I was lucky to receive the teachings of many excellent and inspiring teachers, who have all in some way contributed to driving my curiosity and my desire to learn and study.

Within the ABIBAC section at Lycée Faidherbe, Prof. Brigitte Macrez and Prof. Xavier de Glowczewski have left a lasting impact on me as to the joys of teaching and learning.

At Lycée Louis-le-Grand, I fondly remember Dr. Jill-Jen Vie's (Yorel Reivax!) as well as Dr. Gabriel Scherer's practical classes ('TDs') in Computer Science, who certainly contributed to developing my enjoyment of this field alongside the many discussions with my dear friend and, then, CS mentor Jonathan Laurent.

At ENS Paris-Saclay, I was blessed with a thorough and highly interesting introduction to Computer Science's many subdomains. In particular, Prof. David Baelde, Prof. Serge Haddad and Prof. Hélène Windish have proved time and time again to be both outstanding teachers and trustworthy, supportive human-beings.

At IRCAM, I found a friendly and exciting environment, both during the Acoustique, Traitement du Signal et Informatique Appliqués à la Musique (ATIAM) master's degree as well as during my own PhD. I sincerely thank all the professors whose teaching I received during ATIAM for the many insights they gave me into the vibrant fields of Music Science and Music Technology – long live ATIAM! In particular, I thank Prof. Geoffroy Peeters for his enthusiam, his support and for making me aware of Sony CSL Paris and the music team thereat, when I was in search of directions for the pursuit of my curriculum, effectively kick-starting this whole project. I also thank Dr. Jean-Louis Giavitto for his tremendous support in difficult times and thereafter. I also thank the administrative and technical teams: Sylvie, Cyrielle, Bruno, Patricia, Alexandra and all the others. You are the beating heart of this institution! Special thanks to Eric de Gélis and Bastien Sabarros for making my PhD defense a technically flawless one.

Finally, the MVA Master's degree, through the diversity of the topics it touched on was key to my transition from theoretical CS to statistics

and AI. There, I particularly remember the classes taught by Prof. Stéphane Mallat, Prof. Francis Bach and Prof. Gaël Richard.

I am also thankful to the researchers I had the opportunity of pursuing research internships with: Prof. David Janin at the LaBRI – Université de Bordeaux, Pr Shlomo Dubnov at the CALIT2 – UC San Diego, Prof. Philippe Esling at IRCAM, Prof. François Pachet and Dr. Mathieu Ramona at Sony CSL Paris. These all shaped my scientific approach and it is both a privilege and a pleasure to have gotten to work on so many problems and approaches prior to starting my own research.

Many thanks to my dear friend Dr. Manu Pariente and to Prof. Daniel Pressnitzer for giving me an opportunity to work with them in the pleasant environment of the LSP.

I wholeheartedly thank Emmanuel Deruty, Dr. Gaëtan Hadjeres, Dr. Matthias Demoucron and Prof. Vittorio Loreto at Sony CSL Paris for accepting me back into the music team after my time at Spotify and for making all this journey possible.

I finally thank Virgile Boulanger for letting me into his team at MWM, giving me the chance to finish writing up this manuscript on top of providing me with a delightful way of honing the skills I got to develop during this PhD.

As a side-note, a ton of special thanks to Yotam Mann for making `Tone.js`, a crucial block for all the tools presented in this manuscript, as well for as the inspiration he has provided me with through his work when I was starting out with web development.

SOME MARVELOUS FRIENDS First and foremost, praises are due to my family, Odile, Franck, Lola, Billie and Weti, you mean the world to me and I would not be the person I am if not for you. Also, praises to Grototo, I am eager to see how your journey unfolds.

I thank from the bottom of my heart all my friends at Sony CSL for making it such a stimulating and welcoming environment: Akama-san, Alain, Alexis, Alienor, Amaury, Cristina, Cyran, David, Emmanuel, Gaëtan, Giulio, Inès, Javier, Kishi-san, Léopold, Martina, Matthias, Michael A., Michael T., Natsume-san, Natsumi-sensei, Peter, Pietro, Pratik, Sophie, Stefan, Thomas, Vittorio. There have been hardships, but this lab really feels like a family and you are all significant members of it. Also the extended SONIFY family: Alex, Émilie, Juanjo, Marco, Mathieu, Vincent!

My friends from IRCAM: Alice, Adrien B., Adrien Y., Antoine, Axel, Charles, Clément, Constance, Gabriel, Hadrien, Léo S., Luc, Maxime, Mathieu, Mickaël, Nadia, Tristan, Victor, Vincent Martin, Vincent Martos, Yann.

Many, many, many thanks to Léonard Blier and Jonathan Laurent. Your inspiration and guidance have in many ways been shaping my academic path and my life for more than a decade!

Many friends have supported me in various ways along the way, and I'm so lucky to know all of you. I may forget some names here, so feel free to shoot me a message if you read this, don't find your name and feel vexed, I will gladly offer you a drink to make up for it! Here goes, though: Adrien B., Albane J., Alexandre P., Aline G., Ana-Sofia A. A., Anna-Gaëlle L., Anouk G., Antoine W., Aurélien B., Barnabé F., Bertrand D., Camille R., Cecilia de A., Charlie J., Clara C., Clara D., Clément B., Clément M., Constance B., Constance D., Félicien C., Élie M., Jean-Christophe V., Jeanne G., Laurent B., Léonore S., Margo S., Maria R., Mariam M., Marie D., Marine D., Mathias S.-M., Mathilde S., Matthieu P., Mehdi R., m4dz, Mélodie C., Merwan M., Nath M., Nicolas P., Nina C., Pierre-Léo B., Quentin D., Raquel D.N., Ryadh A., Stéphane O., Théo G., Théo W., Valentin de B., Victor R., Vincent M., Vincent D., Yacine S., Yupanqui R.

Many thanks to all the (extended) Boucly family for welcoming me so often: Alice, Aurélien, Camille, Émilie, Julien, Lucile, Mathilde, Michel, Pierre, Quentin, Suzelle, Véronique, Victoria.

Above all, thanks to Quitterie for your love, your friendship, your attention, your care, your presence, your patience, your smiles, your thoughts. Getting to know you has taught me more than any PhD will certainly ever teach me. None of this would be here if not for you. You are better than marmalade. You are life to me.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ADSR Attack Decay Sustain Release

ARA Audio Random Access

ASAP AudioSculpt As Plugins

ATIAM Acoustique, Traitement du Signal et Informatique Appliqués à la Musique

AU Audio Unit

CLI Command-Line Interface

CSI Creativity Support Index

CUDA Compute Unified Device Architecture

DAW Digital Audio Workstation

DIY Do It Yourself

DDSP Differentiable Digital Signal Processing

FX effect

GAN Generative Adversarial Network

GPU Graphical Processing Unit

GNU GPL GNU General Public License

HCI Human-Computer Interaction

IF Instantaneous Frequency

IRCAM Institut de Recherche et de Coordination Acoustique-Musique

IML Interactive Machine Learning

IPC Inter-Process Communication

ISMIR International Symposium on Music Information Retrieval

LSTM Long Short-Term Memory

ML Machine Learning

NIME New Interfaces for Musical Expression

NN Neural Network

OSC Open Sound Control

OSMD Open Sheet Music Display

RAVE Realtime Audio Variational Encoder

TISMIR Transactions of the International Society for Music Information Retrieval

UI User Interface

UX User Experience

VAE Variational Auto-Encoder

VST Virtual Studio Technology

VQ Vector-Quantization

VQ-CPC Vector-Quantized Contrastive Predictive Coding

VQ-VAE Vector-Quantized Variational Auto-Encoder

1 INTRODUCTION

At its very essence, music deals with the structuring of time in an inherently complex, multiscale and hierarchic view: from the sub-millisecond vibrations of a string to operas lasting several hours, from the few seconds of a melodic line to its resurgence as an ostinato theme throughout a piece or a movie, from a single bass sound to its near-infinite echos throughout a dub track via delay-based processing, from the sound of a synthetic drum machine to decades of inspiration in electronic and hip-hop music. Admittedly, most classical Western music theory can be seen as an endeavor into providing an *effective conceptual structure to time* so that it could be conveniently manipulated for musical creation. This involved the development of several, successive layers of abstraction: from individual *notes* to *melodies* and *harmony*, with *counterpoint* appearing when combining several instruments in parallel, themselves assembled into whole *movements* with fixed forms and typical grammar-like articulations... Much as human natural language deals with the structured organization of meaning, music is a language of time and frequencies.

Accordingly, modern music production is a complex cultural and technical process, which involves knowledge and mastery of multiple specific practices, each focusing on different aspects and different levels of this multiscale process. In this perspective, instrument-playing skills or digital sound design and processing help to sculpt the very fabric of vibrations at microscales, whereas music theory (with scopes potentially much wider than only Western music theory) expertise on melodic, rhythmic and harmonic construction enable the introduction of seconds-scale structure and arrangement and compositional process open up to higher-scale organization of multiple individual instruments and performers [27]. This separation of concerns into specific tasks highlights how music creation has evolved to allow musicians to efficiently operate *across* its multiple time-scales of interest.

With the advent of computer-based music production and the apparition of Digital Audio Workstations (DAWs), all of these practices can now be tackled on a single, consumer laptop computer. Yet this should not at all come for granted: designing proper user interfaces

that address the breadth of the multiple temporal scales in music creation is highly challenging, and has necessitated the development of many abstractions. *Controllers*, such as keyboards, step-sequencers and piano-rolls, allow playing, recording and editing sequences of notes ; synthesizers have been developed as a convenient, buttons- and knobs-based abstraction to control generative processes of physical sound ; interactive pattern-based timeline views enable the organization of multiple such sequences to arrange and form long-form pieces.

One such abstraction – of importance as we enter this work – is the distinction between *symbolic* music and *signal*-level representations or, to put it more simply, *audio*. Symbolic music generally describes the *music-theoretic* view of music in an abstract form as sequences of notes organized in time through rhythm, melodic-construction and harmony. In computer terms, it amounts to compact representations which only describe music pieces as a high-level collection of abstract events, such as playing a given note on a given (abstract) instrument or pressing and releasing a sustain pedal on an (abstract) piano. Crucially, these representations – typically derived from Western notions of sheet-based music representation – discard any aspects of sound texture and the physicality of sound. These aspects are left to audio-based representation, which in technical terms correspond to the recording of sounds either analogically on mediums such as magnetic tapes or records, or numerically by means of digital sampling. These representations are typically capable of capturing infinitely finer details of the music, at the natural expense of being much more demanding in terms of representation dimensionality. If a whole opera can be compactly represented as a series of a few thousands of notes and the whole sheet music for Wagner's opera *Siegfried*, lasting approximately four hours, can be represented over just 439 pages of human-readable content[1], consumer-grade audio signals are typically sampled at a frequency of at least 44,1kHz, that is, one second of audio signal corresponds to 44100 individual numerical values. This makes audio representations much more challenging to use, albeit much more rich and flexible – and much less *biased*.

Indeed, the musical abstractions described in the previous paragraphs are not void of assumptions and generally incorporate various, culturally specific notions of musical creation. I argue in this thesis that these abstractions should constantly be challenged, if only because conceptual and technical innovations in the means of producing

1 https://www.sheetmusicplus.com/title/siegfried-sheet-music/3494848

music have often directly contributed to advancements in the art of music itself. Examples of this abound, with the advent of synthesizers bringing with it genres such as Krautrock and New Age music first, then Techno, House and countless other genres, ultimately becoming ubiquitous in modern music. Similarly, drum machines and samplers paved the way for Hip-Hop and much of electronic music alike and the invention (and subsequent intentional misuse) of vocal effect *Auto-Tune* by American company *Antares Audio Technologies* in 1996 opened up to radically new perceptions of the affects of the human voice and an expansion of its musical possibilities [91]. Rightfully so, one can wonder: are current abstractions for representing and interacting with music ideal? If not, how could we expand on or diverge from them? In asking these initial questions, several viable criticisms quickly emerge, which intertwine technical, æsthetic, even political, concerns.

As a very general motivation first, I note that with the aforementioned vast diversity of tasks at hand in modern music creation, music producers could very well benefit from novel means of creative tools. Those could enable either more *effective* operations on repetitive tasks or tasks otherwise requiring very specific expert skills that they maybe do not master, or they could actually offer radically new aesthetics, expanding on the creative palette of musicians.

On a technical level, second, most of the modern software interfaces for music creation have actually been developed over the course of the 20th century, initially as hardware machines and subsequently transformed into software, often through direct emulation of their original hardware counterpart. Are these metaphors – many of them dating from the 70s or 80s – still up-to-date? [33] In particular, as detailed in Chapter 2 through an overview of classic software metaphors for music recording and music creation, the multiplicity of time-scales involved in music production and the high-precision required at each of those scales makes for interfaces that are sufficiently though disputably effective on desktop computers although those benefit from precise, mouse-based input capabilities. Yet these same interfaces become arguably impractical on the small, low-precision touch-screens of modern mobile devices. As a result of this, most interfaces for mobile music creation tend to be very simplified versions of the desktop counterparts, with only limited creative capacities.

Computers however tend to wane in availability and usage these days in favor of mobile devices, and especially so in developing countries. This could become a cultural issue, as people might progressively lose straightforward access to effective means of producing music, which could impede serendipitous encounters with music creation tools, beneficial for the emergence of new artists. Ultimately, it would feel like a missed opportunity for global creativity to not try and offer convincing tools for music creation on the devices that people *already own* rather than requiring them to first acquire specific hardware to then only be able to start creating some music. This is further discussed in Section 2.1.1.

Finally, as discussed with more details in Chapter 2, the aforementioned abstractions, which arguably underlie the vast majority of current professional music creation software, are far from being *inclusive*, in that they largely hinder the creation of music which does not adhere to the Western template. In a piece from 2021 entitled *Decolonizing Electronic Music Starts With Its Software* for online music publication *Pitchfork*, author Tom Faber writes [40]:

> Unassuming as they may seem, these technologies are far from neutral. Like social media platforms, dating apps, and all data-driven algorithms, music production tools have the unconscious biases of their creators baked into their architecture. If a musician opens a new composition and they are given a 4/4 beat[2] and equal tempered tuning[3] by default, it is implied that other musical systems do not exist, or at least that they are of less value.

Researcher and blogger Jon Silpayamanant, who shares the above quote on his blog, adds [101]:

> As I and many other BIPOC[4] creators have long complained, we'd get more composing or creating done if we didn't have to constantly find workarounds for the musical styles, genres, systems that we prefer working in rather than the default Western and Eurocentric DAW.

The question I ask in this thesis is therefore: can we design novel representations and metaphors of music, better adapted to interactive creation and to modern technological and cultural practices? To be successful and have a chance of finding adoption, these tools should

2 The basic rhythmic structure typical of the vast majority of Western popular music.
3 The ubiquitous scale of musical frequencies and notes used in Western music.
4 Black, Indigenous and People of Color

enable the creation of rich, expressive music in a way that would be accessible to *novice users* whilst also being *powerful enough* to not limit the creativity of expert users. This second aspect could be judged by these tools being of appeal to *existing professional musicians*, arguably very relevant judges in this context. Additionally, we would like these novel interactive representations to be up-to-date and relevant with respect to current technical and cultural practices: on a technical level, they should be able to operate well on modern mobile devices, on a cultural level, they should be ready for adaptation to music creation in non-Western practices.

This is obviously a challenging endeavor: we are requiring these tools to make it possible to create music with both expressive, fine-grained details *and* large-scale compositional structures whilst being usable on mobile devices offering very coarse input capabilities. On top of that, the fine-grained structures themselves (micro-timings, tuning/pitch scales...) would be likely to dramatically change from user to user, switching from one cultural context to another. These tools should ultimately strike an appropriate balance between fine-grained *scales of representation* on one side, enabling to represent fine-grained, expressive details, and a human-interaction-compatible *scale of interactive creation* on the other side. One could very well argue here that there may not be a "one-size fits all" solution to these multiple challenges.

I suggest in this thesis that modern, deep generative statistical models of music can provide convincing, novel approaches to tackling these issues. Indeed, deep learning-based approaches are famed for their ability to learn *higher-level representations* of the data they are trained on [46, 83], representations which I posit in this PhD can be engineered and leveraged *specifically* for the design of novel, *interactive* scales of musical creation.

The idea of applying AI tools to music creation is not new, and the field of generative AI for music is vibrant already. These models have been shown to be able to generate convincing piano performances [49, 59], Irish and Scandinavian folk music [51, 103], and have even recently been scaled to generating full, polyphonic songs with original vocals in many genres [1, 12, 28]. Nevertheless, although generative AI models have received tremendous public attention in recent months in their application to both dialogue – with the ChatGPT model [85] – and image generation – with e.g. the DALL·E 2 and Stable Diffusion

models [94] –, such wide-spread public enthusiasm and adoption is not yet to be seen with AI-assisted music creation tools. I analyze potential reasons for this in Section 3.3.

Somewhat unsurprisingly, one may add.

In short though, I argue that interaction in statistical models does not come for free. Quite the contrary: interaction should actually be directly built into the very structure of the models developed, if those are ultimately to be useful to users and if one wants to properly harness the generative capacity of these models. Arguably, in the case of music, much focus has been devoted to the development of powerful models for varied use cases, but not so much to the development of appropriate, effective user interfaces for those models, as discussed in a 2021 Master's degree thesis by Rebecca Leger entitled *Schnittstellen zur interaktiven Nutzung von Künstlicher Intelligenz in der Musikproduktion*[5]. Ultimately, by making these concerns an afterthought, interactive aspects are hard to recover down the road.

Indeed, in a seminal work by Theis, Oord, and Bethge on the *quantitative* evaluation of generative models, the authors state that such evaluation in an *unambiguous* way is hard and suggest that "generative models need to be evaluated directly with respect to the application(s) they were intended for" [107]. Not doing so puts the proposed approaches at risk of ultimately missing their target, by not being properly adapted to their intended use case(s). In this perspective, in User Experience (UX) terms, the lack of work on integrating those technologies into effective and easy-to-use User Interfaces (UIs) limits their usability by musicians, which in turn dramatically limits the possibility of evaluating their use in *ecologically* relevant settings [18], that is, the setting in which a given tool is ultimately meant to be used. As it stands, musicians are likely to be curious of novel tools and practices and are very often experts in the usage of DAWs but cannot reasonably be expected in all generality to exhibit the near-graduate-level Computer Science literacy often required to even initiate usage of many creative tools developed in music AI research. These requirements for ease-of-use are even more stringent when the tools should additionally be usable by non-experts such as young children or music production novices. Here, designing usable and accessible modalities of interacting with the proposed technologies is essential to getting meaningful responses and feedback from these users.

5 To be published. The title translates to *On user interfaces for the interactive usage of artificial intelligence in music production* [70] and the thesis offers a wide-ranging, systematic literature review of existing interactive applications of AI for music creation.

A very welcome example, nevertheless, of the interest in the music AI community for such "ecological" evaluation is the AI *Song Contest*. This international contest has been held yearly since 2020 and offers to "explore how humans can make music in collaboration with artificial intelligence", by pitting against each other international teams of musicians, researchers and developers in a Eurovision-inspired music creation challenge. Teams are tasked with creating a 4-minutes-long song involving AI in one way or another and are doubly evaluated, first by public vote, then by a jury of scientific and musical domain experts. By asking each participating team to provide a *process document* describing the way AI was used in the creative process, these competitions have been providing a trove of insights on the way musicians can already make use of these techniques in real-world music contexts. These findings have been described in a paper by the contest's organizers after the first edition [58]. In this insightful paper the authors note that teams were able to demonstrate many innovative processes, but the authors also list numerous challenges commonly faced by these users when trying to apply existing AI solutions for music creation, noting:

> As ML models are not easily steerable, teams also generated massive numbers of samples and curated them post-hoc, or used a range of strategies to direct the generation, or algorithmically ranked the samples. Ultimately, teams not only had to manage the "flare and focus" aspects of the creative process, but also juggle them with a parallel process of exploring and curating multiple ML models and outputs. These findings reflect a need to design machine learning-powered music interfaces that are more decomposable, steerable, interpretable, and adaptive, which in return will enable artists to more effectively explore how AI can extend their personal expression.

In the case of designing novel models and metaphors for music creation, this therefore calls for *participatory design*: if the intended use case for the approaches developed is for them to be used by humans (professionals or not) to create music, then the gold standard for the success of these models should be humans actually attempting to make music with those models. Saying this, however, does not solve the problem, as Prof. Wendy Mackay observes in a discussion with Prof. Marcelo M. Wanderley on her work at the frontier of Human-Computer Interaction (HCI), music and art [111]. In this interview, she discusses common flaws she perceives in the evaluation of novel "creative" HCI tools and technologies in academic papers:

> *Mackay.* – I am not really a fan of "Let's evaluate how creative this was or how good this piece was" because I do not think ...
>
> *Wanderley.* – "*Ten graduate students from our lab took part in the experiment...*"
>
> *Mackay.* – Yes, exactly. This only works if the goal is to increase performance, not to provide creative tools for creative professionals. You end up pretending to do something that is not real.

In the same interview, Mackay adds:

> Formal experiments tend to focus on whether or not it does the thing we said it was going to do... [...] Yes, you can measure that, but I think it is more interesting to explore how far afield people were able to go. How much did they explore and do new things that neither you nor they expected? That is also a valuable contribution and a legitimate thing to evaluate. Researchers evaluate both the technology and the human side, usually in terms of their preferences. I prefer to focus on the details of the interaction in a particular context, according to three principles:
>
> - Is it discoverable?
> - Is it appropriable?
> - Is it expressive?

Mackay defines *discoverability* as the ability for people to understand what the proposed tool does and how to operate it without explanations, *appropriability* as the ability for users to *adapt* the technology to their own needs and ways. Finally, she discusses *expressiveness* in instruments as actually being a reflection of the (potentially unexpected) ways in which *people interact with the tool*, much more than an autonomous notion of the *instrument's* expressivity. This interview ends with what can be held as a research program:

> I think problem-finding is harder than problem-solving. If you find a good problem, solving it is usually pretty straightforward. The trick is how to articulate an interesting new design problem.

1.1 DESIGN APPROACH

Following along the lines of this interview, I try and identify key aspects of interactivity in music creation, derived from an analysis of existing, non-AI-assisted classical UIs for music creation and their limitations, as well as from existing proposals for AI-assisted music creation. For this PhD, I specifically posit the following hypotheses:

ON THE IMPORTANCE OF SCALE *Scale* (in both time and frequency) is a quintessential aspect of musical creation and should be addressed as such in the design of music creation tools. The current scales at which music is represented may be at odds with the design of effective user interfaces: although existing representations are generally accurate and convenient visualizations of the relevant aspects of the musical objects they depict, their high precision makes for cumbersome, complex interactions. Accordingly, it may be beneficial to separate the concerns of *visualization* and of *creative interaction*. This separation of concerns represents a conceptual shift from the current ubiquitous use of musical representations *identically* for both visualization and interaction. As such, *designing and engineering* novel scales of musical representation *for* interaction can bring with itself novel, unprecedented modalities of interaction with the potential to break away from current physical and conceptual limitations of existing metaphors.

DESIGNING NOVEL SCALES OF MUSIC WITH AI Modern, *hierarchical* and *multiscale* statistical models of music enable the design of such novel scales of music representation. These representations, typically obtained via *encoder-decoder* architectures, offer an intermediate level between the existing micro and macro scales of representation, whilst staying able to model and capture fine-grained, microscale details of music. Practically, these AI-driven representations operate by slicing the musical material of interest into intermediate-scaled sub-zones. These zones, much alike *puzzle pieces* that can be freely recombined, can then be manipulated themselves as higher-level, conceptual objects.

Examples of these intermediate puzzle pieces in the context of this PhD are:

- "*a slice of one quarter note of a sheet, across four voices*" in the larger context of a four-part chorale in the style of Bach.
  - Here, puzzle pieces, potentially containing multiple notes across multiple voices, would intuitively describe the tonal and rhythmic content of the corresponding zones. There would be puzzle pieces for song beginnings, pieces for song endings, and so forth, with each of these also having multiple variations: fast beginnings, slow beginnings, sad beginnings, simple beginnings, complex beginnings...

- "*one time-frequency atom of duration one second and frequency-width* 200Hz" in the larger context of time-frequency representations of instrument sounds.

These mid-scale puzzle pieces can then be conveniently assembled in time and frequency into large-scale puzzles to recreate the full picture of the objects they represent. Conversely, these statistically-learned puzzle pieces retain fine-grained details of the original musical material and the resulting puzzle-like representations can be *decoded* back to the original, full-resolution musical objects. This is precious, since this means that these models can be made to preserve salient, culturally relevant aspects of expressivity present in the music and sounds they are trained on.

INTERACTION BY DESIGN The resulting representations can furthermore be made *interactive*. This can be seen as playing a guessing game with an additional, auxiliary AI-model, trained specifically to "play" with the aforementioned AI-driven puzzle pieces, once these have been learned. In this game, the user takes a puzzle-like representation of a musical object obtained through the previously trained AI-driven encoder and selectively removes one or more pieces that they do not consider fit, then asks the auxiliary model to replace the resulting blanks with other pieces, whilst accounting for the other puzzle pieces still in place. The idea is therefore to perform *local* updates, with each replaced puzzle piece describing a different realization for the zone in question, whilst maintaining a general sense of homogeneity and cohesion in the resulting, larger-scale material. Provided that the size of the individual pieces be properly devised and adapted to interaction – typically, these will be sized in order to enable convenient access and manipulation even on small touchscreen –, these puzzle-like scales therefore provide a direct, convenient and intuitive higher-level access to low-level, fine-grained aspects of sounds and compositions.

THE RESULTING INTERACTIONS ARE NATURAL Indeed, since these representations essentially share the same spatio-temporal structure as the source material, albeit at a slightly coarser, interaction-adapted scale, they can be *naturally* displayed as *interactive overlays over* the original representations of music for each relevant case. This is highly beneficial, since these original representations are objects that people (or at the very least musicians) are usually familiar with, which ensures that musically-literate users can directly transfer and benefit from

their ability to read and analyze these representations, enabling the *discoverability* of the resulting interfaces by these users.

Such interactive overlays thus enable a novel "point-and-click" approach to music creation, wherein users can simply select zones on this mid-level representation to transform the underlying musical content. This effectively turns common *visualizations* of sound and music into immediate, easy-to-grasp *interactive instruments* with a simple yet rich *one-to-many* [60] interaction modality, wherein a simple click of the user can e. g. trigger the generation of *multiple* notes in the selected range of a co-created symbolic sequence of notes or the localized introduction of rich timbre textures in a co-created instrument sound.

... AND THEY ARE ACCESSIBLE TO NEWCOMERS The shift to a range selection-based approach however also enables effective and immediate usage of these interface by non-experts, since each interactive operation only amounts to *visually* selecting zones to indicate to the generative model which portion of the musical material should be regenerated. *Localization* is therefore the key and arguably single salient aspect of these interactions, so that users can know at any time which positions on the representation correspond to the portions they want to regenerate. By integrating clear and highly contrasted *visual cues* in the interface, synchronized with the playback of the musical object currently being created (a sound, a sequence of notes...), we can help users, irrespectively of their pre-existing musical literacy, locate themselves easily and in an *accessible* way on these representations. Such cues include enlarging, highlighting and coloring interactive zones of the representation whenever the playback head is passing through them. Thanks to the resulting time-aligned audiovisual mapping, users can then locate – by ear and sight – which zones they feel would benefit from transformation, and simply click on those to obtain a new proposal for this region. This ultimately shifts the creative process from one commonly relying on multiple aspects of music theory to one *purely driven by the user's ear and their intrinsic musical tastes*.

THESE REPRESENTATIONS ARE FLUID AND FLEXIBLE The flexibility of modern AI models in terms of analysis and synthesis enables fluid transitioning between several representations of sound and music. This promotes *appropriability* of these tools by making it possible for user to easily use them with their *own data* (sounds or note sequences).

I for instance show how a model trained for the interactive generation of symbolic representations of piano performances can be integrated within an intuitive UI as a tool that enables users to operate both on their own performances (by recording themselves directly into the interface via hardware MIDI keyboards) or even on audio recordings of musical performances, thanks to recent models of audio piano music transcription. Since the underlying auxiliary generative models for interactive transformation account for the current larger-scale context (the whole "puzzle") of the generated material when performing local re-generations, the ability to import data into these tools allows users to guide these generative processes *at inference time* with their own data, that is, *their own style and aesthetics*.

THESE INTERMEDIATE REPRESENTATIONS ARE COMPATIBLE WITH DIVERSE MUSICAL PRACTICES I furthermore hypothesize – although this for now remains unimplemented – that the interaction modalities offered by the interfaces I propose are essentially independent of the exact scale and structure of the underlying, lower-level representation. This makes them in principle perfectly compatible with representations that would for instance not rely on Western pitch or on fixed, metronomic rhythmic grids. In particular, I posit that the novel mid-level representation used for the PIANOTO piano-performance creation interface I propose in Chapter 5 is essentially independent of the Western 12-tone pitch-scale. Indeed, this interface focuses only on a free-from notion of time and offers an interaction modality where users select time slices to regenerate and the AI model generates contextually coherent notes inside those sections, with expressive timing and velocity. As such, it could readily be adapted to the creation of music from non-Western practices in different scales or even with continuous frequencies, as well as with non-quantized rhythms. This could hopefully pave the way to novel advances within the larger, aforementioned discourse on the decolonization of music production tools. As mentioned however, these aspects would still need to be properly evaluated.

Indeed, the ability to generate music from diverse cultural practices and not just Western-adjacent music does not come for free either. It has in particular been shown that not only the choice of training data (obviously of direct importance for the resulting outputs of a generative model) but also the design itself of AI models can entail strong biases and assumptions. In the case of music, these biases essentially

Figure 1.1: The three music creation interfaces I developed as part of this PhD: NONOTO, for sheet music composition, NOTONO for visually-grounded sound creation and PIANOTO, for expressive piano performance creation. Each of these interfaces enables local re-generation of portions of the represented musical material, assisted by a backend AI model.

reflect the very same ones as those underlying common musical representations, making it so that these models generally struggle with the generation of music in non-Western practices. This is discussed at length in Leger's Master degree thesis [70]. Accordingly, and since the core examples this thesis is built upon are indeed themselves directly derived from Western traditions (Bach chorales in Chapter 4, piano performances in Chapter 5 and instrument sounds mostly derived from orchestral instruments in Chapter 7), I want to abstain from too strong and wide-reaching claims. I simply observe that it could be very fruitful to pursue work in exploring those aspects with the interaction concepts proposed in this thesis.

1.2 CONTRIBUTIONS

The above ideas are realized as three different, web-based and open-source music creation prototypes[6], shown on Figure 1.1. I entirely devised, designed, implemented and deployed these three tools as part of the PhD, using modern web development and cloud computing practices. These three music creation tools tackle three essential and widely used representations of music: sheets and piano-rolls for symbolic music first, then time-frequency representations for audio synthesis and sound design – a final prototype for which I also designed, implemented and trained the (equally open-source[7]) underlying AI model. All are designed with the exact same principle, shedding light on the applicability of the concepts described in this

6 Available at github.com/SonyCSLParis/music-inpainting-ts along with demo videos and trained AI models.

7 github.com/SonyCSLParis/interactive-spectrogram-inpainting

introduction. Through the design of AI-driven intermediate layers of representation, I propose to readily transform these *visual* representations into convenient and accessible *interactive instruments*, for usage both by musically-literate users *and* non-musicians alike.

The main AI models used to support NOTONO and PIANOTO, developed at Sony CSL by my advisor Dr. Gaëtan Hadjeres and colleagues, are very much in line with modern AI music generation approaches. I entirely developed, based on ideas by Dr. Gaëtan Hadjeres, the model supporting the NOTONO interface as part of this PhD. This model represented concurrent work in 2020 to the influential JUKEBOX model for audio-based multi-instrument music generation by OPENAI [28]. Interestingly, this simultaneity as well as the proximity of the basic ideas yet the *vastly* different outcomes and use-cases of each approach illustrate one of the essential aspects of my work in the thesis. Both models are indeed based on a similar two-step process. First, one trains an *encoder-decoder model* that embeds densely represented audio-signals into a compact, integer-value multiscale representation – which corresponds to the aforementioned puzzle pieces. Both ours and the JUKEBOX model subsequently train powerful *language models* over these compact representations, to then be able to generate new sounds directly in this compact representation, which can finally be decoded back to a playable audio representation. Such language models, due to their computationally demanding nature, would otherwise be largely intractable on the original, dense audio representation. Yet our approach crucially differs from that of Dhariwal et al. [28] in the very definition of the scale of this puzzle-like intermediate representation – and, consequently, in the nature of the interactions possible with the resulting model.

Whereas JUKEBOX was proposed as a way to tackle large-scale generation of music involving multiple instruments and vocals over durations of tens of seconds, NOTONO focuses on the creation and transformation of individual sounds of a few seconds, and is therefore much more leightweight. As a consequence, JUKEBOX was taking on the order of 8 hours to generate a minute of audio when it was released in 2020 (this latency might have decreased since then thanks to advancements in Machine Learning (ML) hardware and implementations), whereas our model (admittedly also targeting lower resolution) was already operating in real-time on a consumer Graphical Processing Unit (GPU). Accordingly, JUKEBOX, though highly impressive as a technical achievement, offers limited use in interactive settings: its extremely high

computational load and inference times make it difficult to use, especially when one desires to *subsequently transform* the proposed material. This mostly limits its applications to "*generate-then-curate*" approaches, in which users wait for a while to generate a number of different proposals via a model, then are only allowed to decide which ones to keep and which ones to discard, with only limited capabilities for subsequent editing. This approach was applied by musicians-turned-developers duo *Dadabots* as part of their contribution to the 2021 edition of the AI Song Contest. On the page describing their entry, they write[8]:

> We modified OpenAI JUKEBOX to run across a distributed cluster of 36 GPUs, making it easy for us to generate thousands of songs in our sleep with no effort. [...]
>
> We ordered pizza the night we generated this. I ate half, filled out the JSON metadata, queued up the tasks on our 36 GPU cluster, fell asleep, dreaming of being a rockstar. When we woke up we had this song... and a pizza hangover.
>
> Actually that night we had generated 8 albums, 120 songs total, 90-seconds each, of punk rock. They were good, some of them had too much talking, most of them sounded like real songs. Dirty and noisy, lots of screaming, d-beats, choruses, section transitions. We listened through a few of them, this song was pretty good, so we used it. Would've listened to all of them, but we were trying to finish this project last minute in homeroom before class.[9]

This control-free approach, though undoubtedly very fun and punk, arguably represents the exact opposite of what I seek to achieve with the present work.

ACADEMIC PUBLICATIONS All three prototypes have led to first-author academic publications or demos in international conferences:

- NONOTO was presented as an in-person, live demo at NEURIPS 2018 in Montreal, Canada *and* and Industry Track Demo at NEURIPS 2019 in Vancouver, Canada. It was furthermore presented as a conference paper at the 2019 *International Conference on Computational Creativity* (*ICCC 2019*) at UNC Charlotte, US [5] – I presented remotely,
- NOTONO at the 2020 *Joint Conference on AI Music Creativity* [6] – virtual conference, I presented remotely,

8 https://www.aisongcontest.com/participants/dadabots-2021

9 aisongcontest.com/participants/dadabots-2021. This commentary, on the page describing their entry to the contest, should be taken with caution, it may or may not be all truth.

- PIANOTO as a *Late-Breaking Demo* at ISMIR'22 in December 2022 in Bengaluru, India [7] – hybrid conference. I presented and demo'ed PIANOTO remotely (and on Christmas carols, for the occasion).

DISSEMINATION These prototypes have additionally been presented at various international industrial, academic but also mainstream events across:

- *Japan*:
  - NONOTO was presented during the SONY CSL *2019 Open House* in Tokyo – in person demos over three days to members of the industrial and academic sectors as well as journalists. I presented in collaboration with Dr. Gaëtan Hadjeres,
- *Germany*:
  - NONOTO was displayed at the HEINZ-NIXDORF MUSEUMSFORUM FOR COMPUTER SCIENCE in Paderborn – autonomous demo installation which was on display for several weeks in a general-audience-oriented exhibition in Q4 of 2018,
- *Italy*:
  - NOTONO was presented at the 2021 *Rome Maker Faire* – in person demos over three days to a wide audience comprised of representatives from the business, industry and political sectors of Italy as well as young children and students of Rome. I presented in collaboration with Dr. Gaëtan Hadjeres and with the support of SONY CSL's *Technology Promotion Manager* Michael Turbot,
- *France*:
  - *Rencontres Scientifique CIFRE*, with the theme *Artificial Intelligence and Education*, June 8th 2022, discussing the potential application of the techniques and interfaces I propose to musical education with *Messrs.* Axel Jean and Élie Allouche of the Directorate of Digital Education of the French Ministry for National Education and Youth,
  - IRCAM's *Ateliers du Forum* 2022 – in person demos of NOTONO to artists, researchers, students and members of the music technology industry,

– DEEZER FRANCE, I presented NOTONO during a Deezer-internal research seminar in 2022 and PIANOTO as part of Deezer's *Post*-ISMIR *2022 Meetup*,

These presentations and demos were often performed live in conjunction with Ableton Live, in order to showcase these tools in real-world usage contexts, as made possible by the various interoperability modalities I implemented into them. They furthermore covered a diverse range of audiences and provided me every time with some very welcome motivation to try and improve the tools in order to ensure sonically pleasing musical sessions. This thus directly helped the tools mature and become more robust over time. Additionally, these workshops provided me with opportunities of honing the formulation of the concepts underlying these tools, to try and make them accessible to a wide audience. I believe that these concepts actually have the potential to be intuitively understood not only by AI experts, and that proper explanation of these ideas is key to their acceptance by potential users.

These workshops were finally an occasion to evaluate the usability of these tools by non-experts, with generally positive results. Attendees indeed expressed enthusiasm when given access to these tools. Non musically-literate attendees of all ages often noted with joy that they felt like they could create rich musical objects with the prototypes in spite of their limited musical knowledge. Musicians in turn often quickly engaged into discussions on the interconnection modalities of the tools and how they could fit these tools into their workflow, generally expressing interest in getting access to these tools in a mature form.

ARTIST COLLABORATIONS Finally, thanks to multiple collaborations at Sony CSL Paris (with musicians *Uèle Lamore*, *Whim Therapy*, *CHATON*, *DeLaurentis* and *Kadebostany*) and one collaboration outside the lab (with contemporary music duo *YA*, aka *Aline Gorisse* and *Youssra Khechai*), I had the opportunity of having many talented, professional musicians with diverse aesthetic backgrounds use those prototypes for long durations, in particular for the NOTONO prototype. These collaborations directly drove the development of the tools, in a (rather informal however) participatory design approach. I completed this longitudinal study with the administration of a small user survey to evaluate the experience of these expert-users with NOTONO and their relationship with AI-assisted music production.

1.3 OUTLINE

In now detail the general organization of this manuscript.

In order to better situate and contextualize my work, I begin in Chapter 2 with a more in-depth introduction of relevant concepts of music theory, cultural critique and digital music creation, with the aim of highlighting significant limitations of existing interfaces and tools for music creation. Drawing from the various concepts I suggest in this chapter, I then propose in Chapter 3 an overview of existing applications of statistical modeling to music creation, wherein I seek to make a case for the importance of directly addressing the *scale* of interactions. I argue that the lack of focus on interactivity and UIs in existing AI-assisted music creation tools has limited their usability and their adoption. This provides further motivations for the work presented in this manuscript. In the following two parts of the manuscript, I present the three interfaces I proposed, discussing how I believe they provide valuable answers to some limitations of these existing approaches.

More precisely in Part i, I present two interfaces tackling the creation of music in *symbolic* representation: NONOTO, presented in Chapter 4, operates on sheet music, and PIANOTO, presented in Chapter 5, which builds upon recent advances in generative modeling of symbolic data and operates on expressive piano performances. I discuss the respective advantages and downsides of these two prototypes and the use-cases they enable. This discussion focuses particularly on the use of these tools in real-world music production, which has been a key aspect of their design and has hopefully been facilitated by the integration of multiple interconnection modalities, enabling integration into usual music production workflows.

In the second half of this PhD, I then build upon those ideas and propose to approach, under the same general design principles, the *interactive generation of sounds*. This is challenging, since audio data are typically sampled at high-frequency and is therefore much more high-dimensional. In order to make this task tractable, I rely on the Vector-Quantized Variational Auto-Encoder (VQ-VAE) framework, which has previously been applied with success to the generation of sounds, but not under the specific angle of interaction. The concepts underlying the VQ-VAE are introduced in Chapter 6. With these ideas at hand, I then present the NOTONO prototype in Chapter 7. This tool proposes a novel, visual approach to generative sound-design. I finally report, in

Chapter 8, on extensive usage sessions of this AI-driven synthesizer with professional musicians at Sony CSL Paris and outside. These collaborations have enabled a longitudinal study of the tool, driving an artist-centric approach to the design of this tool which has undoubtedly permeated also to the other two tools presented in this work.

Closing this manuscript in Chapter 9, I present a general conclusion to the work, reflecting on the successes and (current) limitations of the proposed approach, as well as future directions for development and research, in the various topics of statistical modeling, interface design and user experience study.

BACKGROUND AND OVERVIEW

2 AN INTRODUCTION TO MUSIC CREATION

In this chapter, I present the general motivations for the work presented in this thesis by proposing a general overview of salient characteristics of modern music creation, mostly focusing on two aspects. First, I recall the diversity of skills at play in producing modern records as well as the diversity of cultural references and musical techniques that musicians juggle with to create music. Here, interactive, generative AI can be (and as previously been) suggested as a means of building statistical models able to capture and subsequently emulate various aesthetics, ultimately offering powerful tools that can support modern existing creative practices and expand them. I then propose an additional, original, motivation, grounded in current considerations on mobile user interface design. Indeed, through an overview of global statistics of electronic device usage I suggest that electronic device users (and especially younger ones) favor mobile devices over desktop computers more and more, and that this is even more true of populations in regions of the world with lower incomes. Yet, most of the existing music production software interfaces have been developed for desktop computers, and actually translate poorly to mobile devices. This can be attributed to historical reasons and a delay in adapting to these more modern devices, but can also be linked to the fact that designing convincing interfaces for professional music production on smartphones is *in itself* challenging, since the high precision involved in music production calls for precise, focused interactions, which are hard to transpose to the small touchscreens of mobile devices. I highlight this through an overview of common music production interfaces in Section 2.2. I argue that a cultural 'soft-lock' could emerge from this situation, with some populations progressively losing access to the relevant tools for digital music production. This serves as an additional motivation for the approach proposed in this PhD, and I argue that AI models offer a solution to design novel interfaces that do not require precise interaction yet enable powerful creation with the ability for long-term planning and rich compositional approaches.

2.1 ON MODERN MUSIC PRODUCTION

Modern music composition is a complex and multi-faceted art: with skills ranging from *vocal or instrumental performance* to *melodic* and *harmonic composition* or *drum patterns sequencing* to *sampling* and *sound synthesis* and back to *mixing*, *recording*, *mastering*… all these aspects and expertise come together to shape the resulting music. As an illustration, the credits to *Random Access Memories*, the 2013 album by French electronic music duo Daft Punk, list more than 100 collaborators in addition to the core duo[1]. Each of these different expert practices is complex and rich enough in its own right to justify full-time professional careers for those who embrace them. With such a wide variety of high-level skills at play to create modern music, appropriate tools and techniques are of the essence. Such tools may enable seasoned professionals to improve on their craft, bettering their results or maybe reducing the fatigue and workload associated to tedious tasks. But these tools should also cater to the needs of independent artists or even hobbyists and help them expand on their creative practice.

BARRIERS TO MUSICAL EXPRESSION In this perspective, *not everybody*[2] can afford studio-grade, professional recording equipment. As such, new tools offering to reduce the literal, *financial* cost of entry to e.g. professional-sounding recording have often had wide emancipating results, such as through the boom of the so-called *laptop musician* and *home-studio* in the 2000s, made possible by the then lowering cost of access to quality recording equipment and stable DAWs finally making these tools widely accessible to hobbyists. Simultaneously, on a theoretical and technical level, *not everybody* knows the rules counterpoint *à-la* J.S. Bach or of total serialism *à-la* K. Stockhausen and P. Boulez. Not everybody, either, masters the craft of record sampling and hip-hop-instrumentals production or of DJing, or of, say, Tibetan throat singing… And *even fewer*[3] people master all of these theories and skills at the same time.

HYBRIDIZATION PAYS Yet, as illustrated by contemporary Spanish artist Rosalía's astounding international success – 12 times Latin Grammy and 1 time Grammy Award winner, 40 million monthly lis-

1 Source: Discogs, discogs.com/master/556257-Daft-Punk-Random-Access-Memories
2 An intentional understatement.
3 Yet another understatement.

teners on the Spotify streaming platform –, incorporating techniques outside of one's typical, expected aesthetics has more than once paved the way to creative breakthroughs and worldwide appraise. In Rosalía's case, her music expertly and respectfully blends the traditional heritage and vocal techniques of Flamenco with the modern, Auto-Tune-driven sounds of Reggaeton and Latin club music.

ON MUSIC TECHNOLOGY Offering tools that lower the *technical* cost of entry to expert techniques could therefore help enable both professionals and hobbyists to integrate them into their work and bring them to new territories. In order to be relevant, such tools or instruments should embrace the aforementioned multi-faceted nature of music: one should not hope to provide "*one-solution fits all*" types of approaches to such a diversity of technical, cultural and aesthetic practices. And with so many challenging aspects, there is without a doubt room for many new such offerings, as can be seen by the variety of topics covered at domain-specific conferences such as New Interfaces for Musical Expression (NIME) or International Symposium on Music Information Retrieval (ISMIR), respectively focusing on the creative aspects of music technology and on the technical aspects of musical creativity.

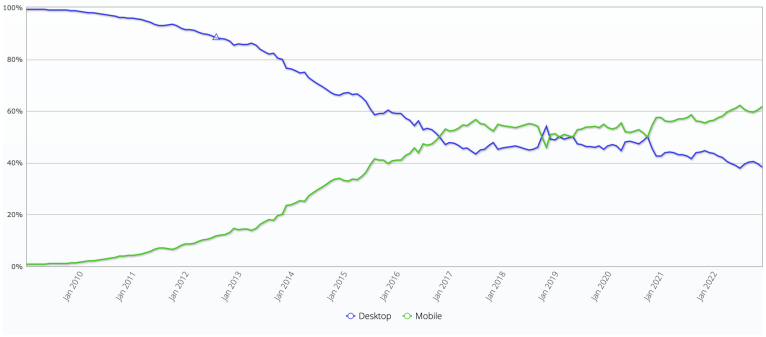
A CASE FOR AI TOOLS Through their ability to learn, via training on relevant datasets, the underlying rules and subtleties of diverse genre of music or types of sounds and subsequently emulate and reproduce them to create novel artifacts, modern AI tools represent an opportunity for the development of innovative tools for creative musical expression. Such AI-based tools could help interested artists get access to diverse, specific aesthetics and techniques in a spontaneous fashion. As I show through a brief literature review in Section 3.3, such ideas have already been widely explored in the last years and our work takes place in the continuity of many prior projects devising applications of statistical modeling to music creation. Yet, I actually go further in this thesis and argue that modern statistical models of music enable the design of radically novel user interfaces which, by enabling the design of novel *scales* of musical interaction, have the potential to significantly widen access to music production. Indeed, in an era where users of electronic devices more and more favor mobile phones over computer (as discussed below), digital music production tools are still overwhelmingly developed for desktops and translate poorly to mobile devices, as discussed in Section 2.2. I posit – though this

is speculative and not based on extensive research – that this could represent a significant cultural issue.

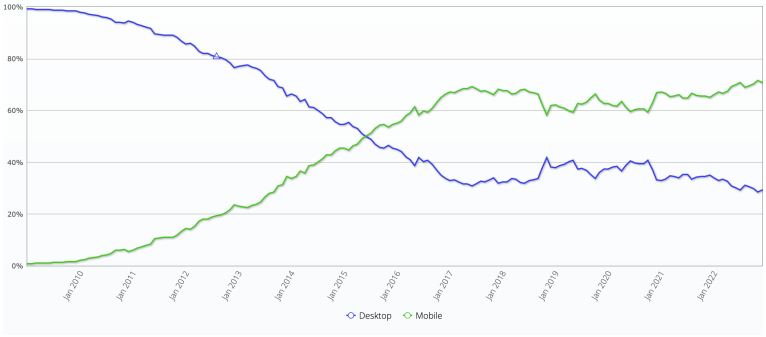
2.1.1 Music production in the smartphone-era

In order to highlight this hypothesis, let us look more closely (with the graphs displayed Figure 2.1) at per-device-type statistics of Internet access, at a worldwide scale as well as more specifically in Africa, Asia and India. These statistics show that, at least when it comes down to accessing the Internet, mobile phones have been steadily growing in adoption over the last decade – and, according to the charts, this imbalance towards mobile devices is still increasing. This is even more true in (broadly speaking) developing countries of the so-called Global South, as visible on the statistics of Africa and of Asia, with India leading the way at 76.64% of market share for mobile devices as of December 2022. In addition, age plays a strong role in these device preferences and younger users tend to also disproportionately favor mobile devices over computers for accessing the Internet, as shown on Figure 2.2. Although there are obviously confounding factors to these stats, such as the simple fact that mobile devices enable Internet access on the go, naturally giving them an obvious advantage in these situations, if we assume that these statistics represent a somewhat reliable proxy to more general device usage and potentially device access, then one can assume that people are progressively shifting away from computers in favor of mobile phones, and that this shift is even more true of younger populations (globally) and of populations in developing countries.

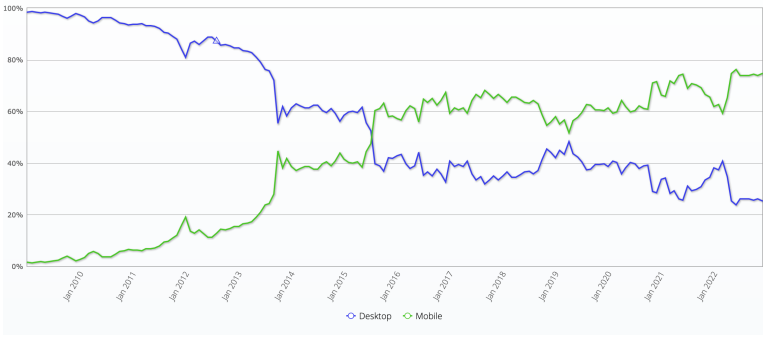
Yet, coming back to the initial topic of music, most professional-grade music production software target desktop computers and strongly rely on the high interaction precision offered by the *mouse + keyboard + large screen* combination, as I underline with more details in Section 2.2. An approach commonly undertaken to circumvent the difficulties of designing convenient interfaces for music production on these small devices has been the design of largely simplified version of existing UIs, intentionally limiting their capacities (such as FL Studio Mobile, or, more recently, Ableton Note). This approach, though it has produced very stimulating tools, often assumes that the users also have access to a computer-based workstation and use the smartphone application to *sketch* ideas, before ultimately moving to the computer to bring these ideas to completion. This is often true of Western, affluent users, but,



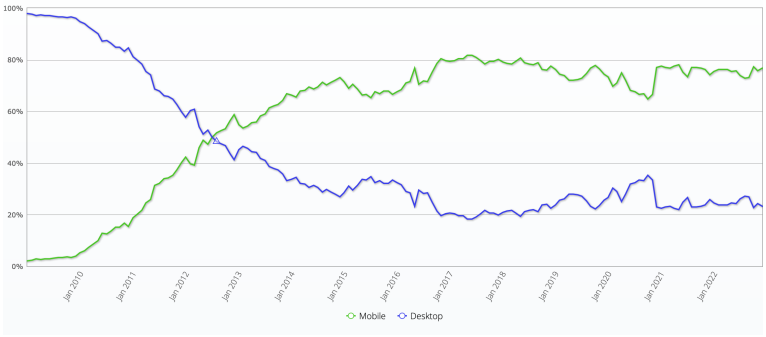
(a) Global (61.42% of mobile usage as of December 2022)



(b) Asia (70.79% of mobile usage as of December 2022)



(c) Africa (74.72% of mobile usage as of December 2022)



(d) India (76.64% of mobile usage as of December 2022)

Figure 2.1: Relative market shares of computer and smartphones for Internet access, January 2009 – December 2022.
Source: statcounter.com. Accessed 2023/01/04.

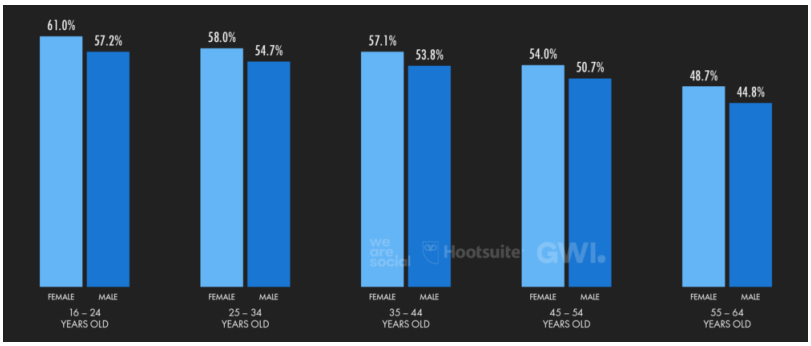


Figure 2.2: Relative market shares of computer and smartphones for Internet access by age and gender, globally, Q1 2022.
Source: We Are Social | The Global State of Digital in July 2022 [65], based on data from the Global Web Index. Accessed 2023/01/07.

as discussed above, can not necessarily be assumed of populations in less wealthy regions of the world. These conditions could become a cultural issue: if no efforts are put into building mobile-ready tools for full-fledged, high-quality music production – that is, not simply musical toys or games, or feature-limited applications –, a large part of the world's population, without regular access to desktop computers, could end up being to some extent locked out of this activity. And, if need be, let us now briefly recast why such a (hypothetical) scenario would represent a significant loss for the worldwide culture, by hinting at the impact that contemporary musical scenes in Africa are having on worldwide popular music.

A DETOUR THROUGH AFRICA'S CONTEMPORARY CLUB MUSIC SCENE

A large part of the contemporary underground African electronic music scene(s) is famed to have been spearheaded by producers aiming from ghettos working on cheap, second-hand computers under cracked versions of the FL Studio DAW [99] (do note, nevertheless, that these musical scenes should by no means be reduced to these socioeconomic conditions, and prominent producers also hailed from the middle and upper classes, as developed by Alisch [2]). These pioneering producers have been blending the sounds of diverse regional African traditions with modern club sounds, giving birth since the 1990s to genres such as Angola's Kuduro[4], South Africa's Gqom[5] and Kwaito[6], or the more extreme Singeli sub-genre, showcasing a

4 Buraka Som Sistema – *Luanda: Lisboa* (2008 – Angola / Portugal)
5 Da Soul Boyz – *Welcome To Durban* (2018 – South Africa / U.K.)
6 Machance & DJ Abbas - *He Kheya Ndon* (2005 – South Africa)

hyper-speed take on traditional Tanzanian rhythms and melodies[7]... These artists have managed to circumvent otherwise oft highly unfavorable material conditions in socioeconomic ghettos to approach music production in a more "Do It Yourself (DIY)" fashion (since typical professional hardware was out of financial reach to most of them). In doing so, they have been driving to a significant extent the vitality of worldwide music. Indeed, these then-niche musical trends are now enjoying mainstream success and visibility through genres such as Amapiano[8], a modern take on House music characterized by the systematic use of a synthesized emulation of traditional South-African wood blocks, or Afrowave. These genres have in turn been inspiring contemporary popular musicians such as on Beyonce's *MY POWER* song from 2018, co-produced by DJ Lag, one of the pioneers of Gqom, which was very positively welcomed, counting more than 35 million streams on Spotify as of January 2023.

This brief detour accounts (though only by scratching the very surface of the topic) for the very significant cultural importance of the musical production originating in Africa – and similar ideas could be developed for many other regions OF THE WORLD, such as Brazil and its *Baile Funk* scene. But if the aforementioned technological divide between desktop-oriented professional music production software and the mobile-favoring populations in these countries was to widen, one could fear that so-to-say serendipitous encounters with music production would wane ; that music enthusiasts, without convenient access to the hardware required to run music production software, would be less likely to engage in a creative practice. Developing tools that are globally accessible to diverse populations, designed to operate on the types of devices that these populations are used to spending time on, should therefore not be an afterthought if music production is to remain widely accessible, preserving the possibility of a vibrant cultural ecosystem.

Again, this is a personal hypothesis and is only an untested scenario, not a proper research statement.

I nevertheless don't mean to overestimate the "necessity" of the work presented here, since, exactly as discussed here, many musicians creating these innovative musical genres have done so in unforeseen, DIY approaches. One should therefore beware of tempting oversimplifications and should not overestimate the dependency of musical scenes in low-income countries on researchers, software designers and developers in affluent countries designing software *for* them: these musicians are more than capable of shaping their own approaches,

7 Bamba Pana – *Biti Three* (2018 – Tanzania)
8 Vigro Deep – *My Rules* (2022 – South Africa)

as discussed by Alisch [2]. Also, obviously, music creation and production has existed in countless forms in these regions for millennia irrespective of the very existence of computers (!). I notwithstanding believe that this represents a stimulating consideration to have in mind as we enter this manuscript.

AI-ASSISTANTS FOR MOBILE-READY USER-INTERFACES Through this PhD I attempt to demonstrate how AI-based tools could help pave the way not only for exciting novel tools for existing musicians but also for more cross-platform tools, enabling access to high-quality music production to mobile-only users. The generative statistical models powering these tools are designed *for* interactive usage and, in turn, enable designing user interfaces which coarsely reduce the need for high-precision interactions, making them usable on mobile devices, whilst providing satisfactory control over the created material and maintaining a high level of creative freedom and generative power, suitable for usage in professional music production situations. The chosen approach is a very practical one: I designed and implemented three novel, cross-platform web-based prototypes that offer new AI-assisted approaches to several typical, *"situated"* music production activities. Two of these prototypes rely on existing AI models and bring these models to life whilst the model for the third prototype was designed and trained as part of the PhD. These prototypes have been validated for their effectiveness for professional music production through extended collaboration with and usage by professional musicians. They have simultaneously been carefully designed to enable usage on mobile *without sacrificing their creative capacity*. In order to better grasp the motivations for the design of these tools and their respective contributions, I now introduce general principles of computer-based music production, as well as more specifically focus on two of the dominant UI models, one for the creation of sounds, the *virtual synthesizer*, the other, the *piano-roll*, for the arrangement of note events. I try to highlight their shortcomings when attempting to transfer them to mobile devices and the resulting tendency in consumer DAWs to only propose over-simplified, almost toy-like versions of these interfaces, strongly limiting their capacities on these devices.

To borrow a term from Hugo Scurto's PhD thesis [97].

2.2 A PRIMER ON MUSIC CREATION

In this section, I introduce various concepts of music production useful for the understanding of the AI models and interfaces discussed in this manuscript. The concepts underlying modern software for digital music production are largely inherited from analog electronic music as pioneered during the 20th century, itself directly influenced by more traditional principles derived from Western, acoustic music composition and performance, based in notions of *written music*. I briefly present these fundamental concepts in Section 2.2.1, then introduce their digital counterparts in Section 2.2.2. But first, a note on musical *scales* and a disclaimer for a limitation of the work presented here.

A SMALL DIGRESSION ON SCALES AND DIVERSITY Note that as part of this thesis, I solely focus on Western notions of music composition. This is mostly visible in the reliance on Western symbolic notations of melody and rhythm and the use of the Western twelve-tone *scale*. A scale describes a collection of musical frequencies which have been grouped together according to æsthetic notions: these frequencies are believed to "work well" with one another. Over centuries of progressive refinement, grounded in notions of vibratory physics and partly driven by convenience in the design of instruments, Western music has converged onto what is considered by many in the West as “the” scale, comprised of the 12 tones of the piano: A, B♭, B, C, C♯ . . . A♭, as shown in music notation on Figure 2.3.

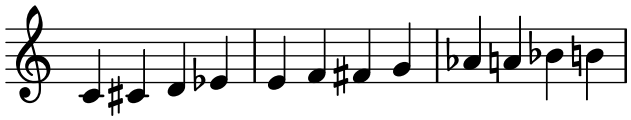


Figure 2.3: The twelve notes from the Western 12-tone scale (from C to B)

Yet this is far from being the sole scale used in music making over the world, and classical Indian music styles such as Hindustani and Carnatic music for instance rely on broadly different scales with more than just 12 subdivisions, with claimed numbers varying between a generally agreed upon 22 sub-intervals or *Shrutis* to up to 96 according to certain music scholars [98, 100]. Yet classical Western notation remains dominant and has served as the basis for the design of most of the tools available today for computed-based music production. These limitations are exhibited in the reliance on discrete, 12 tone keyboards on most hardware synthesizer, limiting their usability for

other types of music. Similarly, most interfaces for the recording and organization of sequences of notes (described with more details in the forthcoming sections) are designed around the 12-tone scale. These limitations are also at play in research on music analysis and creation, and in our case, in current research on AI-assisted interactive tools for music creation. This is discussed by Leger [70]. In her broad literature review of existing models and user interfaces for AI-assisted music creation, Leger – amongst other observations – highlights that most of the proposed solutions focus on and are limited to Western music, both in terms of the music representations and interconnection modalities used, as well as in terms of the datasets used for training the models, largely composed themselves of Western music.

These limitations are equally reflected in the tools I developed as part of this thesis. In particular, the NONOTO interface presented in Chapter 4 is designed to operate over sheet music and directly embodies these concepts. More generally though, NONOTO as well as the other two interfaces presented in this thesis, NOTONO and PIANOTO, currently rely on the twelve-tone scale typical of Western music concepts. This arguably limits the reach of the work presented here, at least in its present form – for instance by forbidding the use of non-western microtonal scales. I nonetheless argue in Chapter 5 that the PIANOTO interface and its interactive modality are actually *independent* of the underlying musical representation used and that this UI element could actually open up the way to a novel, effective way of creating music using non-Western, potentially very rich and fine-grained microtonal scales – provided appropriate interactive AI models trained on these music types. For now, let us nevertheless come back to the original topic of interest.

2.2.1 Elements of music theory

The classical, Western notion of music is rooted in a written tradition, with strong conceptual implications on the way music is approached in these cultures. Traces of written music have been found in Ancient Greece but the modern notions of scales, sheets and the notation used nowadays has been derived first from *neumes*, an early notation used for the writing of liturgical songs [63, 104]. This notation evolved over centuries and was largely fixed by 18th century, although it keeps evolving nowadays [78].

SEPARATING BETWEEN SHEET MUSIC AND PERFORMANCE This tradition separates a *compositional* aspect of music, wherein a composer *defines* the musical content of a piece, such as instrumentation, notes, rhythms, velocities, lyrics... and the actual *performance* of the piece, during which this conceptual description is realized, brought to life by performing musicians. The sound is produced by the musical instruments *controlled* by the musicians. Conceptually, the instruments represent an *interface* onto which musicians can input commands (such as key presses on a piano) to trigger desired notes. The separation between notation and performance has many benefits: musical pieces can be re-played, studied, solely based on the musical sheet, with the potential to outlive their composers by long. Pieces can also be re-arranged, which consists in re-writing the sheet for a different set of instruments. A single musical source can therefore be modified into plenty of different versions for different setups. The reliance on musical notation has also brought with it a tradition of re-interpretation: the same piece can naturally be played several times by different performers, and these multiple interpretations will each bring specific subtleties. Indeed, the actual sonic realization of a piece cannot be entirely prescribed through symbolic notations only, an idea that was brought to an extreme by John Cage with his famous piece 4′33", premiered in Woodstock, New York, on August 29, 1952. The piece is known for being a 'silent' piece, during which the performer should stand silent on stage, not playing a single note. Yet according to the composer, the true intention is to invite listeners to notice and pay attention to the surrounding noise (natural, environmental noise, people coughing or talking...), making each performance of the piece absolutely unique, in spite of the sheet containing no further instructions than the durations for each of the three, silent, movements.

SHEET TIME AND PERFORMANCE TIME A crucial aspect of the difference between a sheet and its rendition lies in the definition of musical time. Indeed, in written notation, note duration and start times, generally denoted as *rhythm*, are defined as multiple and fractions of a reference time unit, the *beat*, as shown on Figure 2.4. This discretized time represents an idealized version of the desired rhythmic content, but any actual interpretation will necessarily drift away from this perfect time. In particular, the rhythmic notations also leave the door open to subjective interpretation by the performer, who can decide to play a given segment faster and slow down for another one. The

real time of a performance is thus an un-quantized, fluid time scale. This distinction between sheet time and performance time is an important one and has technological implications on the different digital representations of symbolic music presented below.

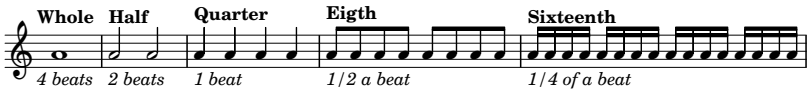


Figure 2.4: Standard note duration values relative to a beat (all measures have the same total duration, highlighting the progressive shortening of the individual rhythmic symbols).

MUSIC AS A MULTI-SCALE PROCESS Equipped with these definitions of musical scales and musical timing, (Western) music composition can be simplified as a *multiscale*, *hierarchical*, *time-frequency process* describing the *temporal evolution of physically vibrating frequencies*. It involves the following elements:

1. At the smallest level (at a scale of seconds), individual notes are arranged in time (through *rhythm*) and frequency (through *scales*, pitches) to create melodic and harmonic parts for individual instrument.
    a) Here, as mentioned before, *two* levels actually co-exist: the fine-grained, continuous physical level of actual, real-world timing and frequencies and the already much more high-level idealized, logical level of discrete pitches and musical rhythm notation.
2. These individual parts are then layered for polyphonic compositions, with multiple instruments and vocalists playing synchronously. The interaction of these multiple, individual lines enables the creation of rich harmonic structures, that is, interplay of several notes (frequencies) at the same timeout, and the temporal organization and progression thereof. Relying on multiple instruments each playing specific parts furthermore enables the creation of rich sound texture by combining heterogeneous sound sources.
3. These layered parts with multiple co-existing instruments create large-scale movements in time (at the scale of tens of seconds to minutes), the organization of which serves to create a logical and symbolic progression throughout the whole piece. An example

of this is the alternance of verse and choruses in songs in popular music, which each different verse and the repeated chorus (and potential variations of it) each represent logical units at a very high-level of the piece.

Drawing from an analogy with models of natural language, individual notes can be seen at the lowest level as either individual characters or words, which are then assembled in succession to create sentences, and those sentences in turn are articulated to tell a full story. This analogy is actually at not just a thought model but constitutes the very basis of many of the models applied in music analysis and generation, as will be seen in the following, wherein models of natural language have been applied with success to music analysis as well as generation [1, 12, 14, 49, 59].

But this simplified process actually holds in itself a blatant over-simplification, lying in the definition of the individual *notes* played or sung by the musicians, which can only hardly be simplified to abstract tokens and do certainly not represent the finest possible scale. Indeed, each of those individual sounds can itself be decomposed and analyzed as a fine-grained temporal evolution of e.g. a physical model, with typical scales ranging from seconds to even micro-seconds. A simple example of such temporal evolution at the note level is in the diversity of amplitude patterns of individual sounds, depending on the types of instruments and on the desired æsthetic characteristics of the produced sound. These amplitude patterns, unfolding at scales of milliseconds, are commonly split into four separate components, which can be seen as yet another abstract language at a sub-note level. These different components can be interpreted according to a physical model of a vibrating body: the *attack* of the sound, during which the object receives an initial excitation. The *decay* part, during which the sound quickly decays as the object reaches a more steady state again. The *sustain* part, in the case of instruments receiving continuous excitation such as wind instruments, where the excitation of the body is maintained in a steady state for a potentially arbitrary duration. Finally, as the excitation stops, the *release* part kicks in and the vibrating body returns to a resting state. Examples of the different types of such Attack Decay Sustain Release (ADSR) patterns for various types of instruments are shown on Figure 2.5. And the timing of these patterns hold much musicality in itself as well.

Professional-grade recordings can be sampled at frequencies of or more 96kHz

Similarly, as mentioned before for rhythm, although musical rhythm could be perceived as a high-level feature adhering to a fixed grid

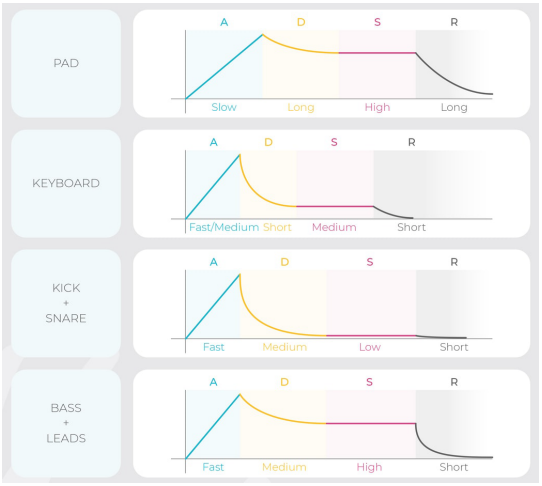


Figure 2.5: Amplitude patterns for various types of instruments (image taken from www.musicianonamission.com)

according to musical notation, tipically ascribing events at the scale of seconds, human musicians inevitably divert from this idealized, abstract representation through micro-level deviations at the scale of milliseconds and this so-called *micro-timing* is often praised exactly for being a crucial component in giving musicality to an interpretation [17, 26, 43]. The same goes for musical frequency, where notes could be seen as the scale of choice, but often only represent a very coarse initial representation, for instance in the analysis of expressive singing voice, and one typically has to move the analysis down to the scale of Hz [27, 98, 110].

ON THE CHALLENGE OF SCALE The analysis (and, similarly, creation) of music therefore represents a difficult task, which typically requires to tackle these multiple, almost fractal, levels of representation, both individually and in their interplay. Individual lines can thus be analyzed for their respective characteristics, focusing here on the individual sound of a specific musician in isolation (low-level) and there on the textual and harmonic evolution throughout a song (higher-level). Yet, the individual-line level, as well as even the physical, micro-level articulations of sounds, can serve a direct function in the more high-level symbolic and logic structure of a piece.

An example of this interplay between low-level, physical affects and high-level symbolic meaning is the series of albums *Everywhere at the End of Time* by English musician *The Caretaker*. From a pure signal-level perspective, this piece, spanning six-and-a-half hours, presents itself as a succession of looped samples of early 20th century ballroom

music, which become progressively more distorted, acoustically degraded and noisy and could be taken solely as such, considering it gratuitously noisy and disturbing music. Symbolically however, this auditory evolution into sonic gray areas serves to suggest and portray the image of a patient suffering from Alzheimer's disease. The piece can therefore be appreciated at a micro-level for the level of detail in the sound design as well as for the individual, often ghostly, melodies appearing in each individual movement. It can however also be embraced at the much larger scale of its whole duration, through the gradual process it seeks to emulate. Both levels of listening ultimately co-exist and are intertwined in the piece.

Appropriate models of music should arguably be able to address these different scales. As far as music creation is concerned, appropriate interfaces for music creation should in turn enable fluid transitioning between those multiple scales to allow users to properly access and control all relevant dimensions of the musical pieces they create. Because however of the high diversity of these scales, from micro-level to macro-level, this is a highly challenging endeavor:

- How is one to compose a full, minutes long piece, if even the articulation of a single sound can be shaped at the scale of milliseconds?
- How is one to effectively operate on such small scales?
- How can the work on these small scales be transported and composed, combined at the higher scales?

In asking those different questions, another set of "scales" naturally arises: that of effective human actions and interactions. Can one physically operate at the microsecond-and-microhertz scale of high-frequency audio signals? Conversely, how can one manipulate very long segments of music to articulate them and build hours-long pieces, such as the previously mentioned piece by *The Caretaker*, without it taking hours for any given minor transformation?

I argue in this thesis that one key component of effective interfaces and metaphors for music creation lies in their ability to provide convenient scales of operation. This is not an essentially new idea: the whole discussion during the previous paragraphs on the multiple scales of written, Western music actually highlights the very same goal in the development of this system! Those notations were introduced as a means to abstract away low-level details of e.g. sonic properties and

enable effective compositional practices for music composers and efficient transmission to performers. But, as discussed, these notations on their own abstract away a lot of the musicality of musical pieces. This is arguably perfectly fine when sheets are used for transmission for human musicians, since they can re-introduce this musicality through interpretation, equipped with a large background of musical knowledge and tradition, but not so much when transmitting those to machines and computers, which will stick to re-playing these cold and inexpressive representations as they are. Additionally, this simplified notation system fails at providing convenient means of action when one wants to operate at the lower-level scales of physical sound.

I therefore propose to use generative, statistical models of music specifically to produce novel, intermediate level scales designed *for* interactive operation, without sacrificing the richness and expressivity of the underlying, more fine-grained content.

SOUND PROCESSING, FXS AND DIVERSIONS THEREOF As an aside and for sake of completion in this description of the core aspects of modern music production, I mention that in addition to this first distinction between the symbolic specification of a piece through e.g. the sheet and its sonic realization by a musician, the development of amplified music in the 20th century has brought with it the concept of effect (FX) chains, which receive audio signal and process it to introduce æsthetic transformations such as echos, reverberation or distortion. Drawing a parallel from the separation described in the previous paragraphs between the sheet, a written, symbolic, abstract representation of music, and the performance, a physical, acoustic realization of it, effects can be seen as units that receive a realization of music in the acoustic domain and produce novel variations of it. Electric guitarists in rock music are well known for the use of rich, sound processing chains to create a custom, powerful sound. But the idea of sound processing is really ubiquitous to all modern music production, whether for *technical* purposes of cleaning and processing audio signals for optimal playback on varied speaker types, or for *creative* usage, coarsely transforming the original signals. Interestingly, the same technologies can often be used for both these goals, depending on the amount of processing performed. A famous example is the AutoTune vocal processing software: initially conceived in 1997 to allow post-processing of recorded vocals to correct vocal imperfections and wrong, off-key notes through slight transformations, it was

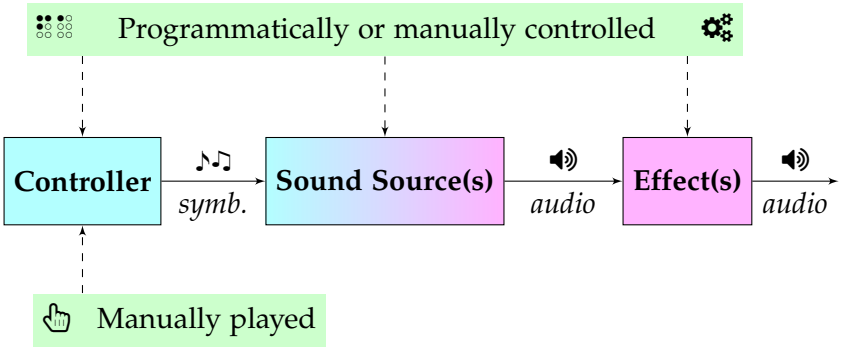


Figure 2.6: Abstract representation of the typical control and data flow within DAWs. Blue depicts symbolic data and pink denotes audio data. Green represents possible programmatic or manual control.

quickly diverted from this intent to instead produce stark, unnatural sounding vocal transformations as pioneered in Cher's 1998 song, *Believe*. This effect has seen been widely used across a many genres in contemporary pop music, as highlighted by influential British music critic Simon Reynolds [91]. This detour additionally highlights a key aspect of many music creation tools, which are often diverted from their originally intended use-case by end-users. Such a tendency to diversion can serve as a motivation when designing novel music creation tools and interfaces to make sure that these tools be as *opinionated* (helping them outline a unique identity) as they be *flexible*, ensuring that they can also be bent to unplanned-for usage.

2.2.2 Principles of Digital Audio Workstations

The separation between a conceptual, *symbolic*, notes-related side and its physical *audio* realization is actually one of the pillars of modern digital (both hardware or software) music production, which enables a very convenient, *modular* use of software and hardware interfaces. Indeed most popular software workstations for music production are based on a tripartite separation of concerns between: *controllers*, *sound sources*, and *effects*, describing the common information flow in most usual DAWs, as represented on Figure 2.6 [33].

Broadly speaking, controllers emit abstract commands. These commands typically describe notes to play or mute and how (e.g. how loud) to play those. In turn, these controller-emitted commands activate sound sources that produce tones and sounds or play back pre-recorded audio samples. Finally, the heterogeneous audio signals produced by the various sound sources in presence are (optionally)

processed individually or together via effects, usually denoted as FXs. This separation is very much the same as the one underlying sheet music in traditional, non-computer music: the controller represents the sheet or the physical control interface of an instrument (such as a piano's keyboard, or the strings on a violin), used to describe the notes to play, along with additional high-level indications of technique or aesthetic intent, which are then rendered by the sound sources which are physical body of the musician's instruments. Through this decoupled representation and the use of interoperable standard technologies, a single controller playing a programmed note sequence can be used to trigger multiple sound sources simultaneously. Similarly, sound sources as well as effects can be hot-swapped during playback, enabling to create complex routings of signals, yielding infinite possibilities for sound synthesis and playback. More crucially, separating notes from signals enables *iterative compositional approaches*: one can record some notes (at the symbolic level) through a controller, then play them back with a synthesizer, only to realize afterwards that one of those notes was misplaced or out of tune. Thanks to the decoupled representation, such issues can be fixed simply by editing the note in question directly from the symbolic representation separately from any audio rendering aspects. Otherwise, one would need to re-record the whole sequence, or at least re-record the failed portion and manually re-insert it into the recording by means of collage (as used to be done when recording with physical, analog hardware such as magnetic tapes).

Note that the representation in Figure 2.6 is a simplification. In reality, it is customary to have *multiple*, distinct controllers feeding into *multiple*, distinct sound sources, with any given controller possibly targeting *multiple* sound sources at the same time. Each of these sound sources can then receive an arbitrary number of chained, indivual effect units. This results in multiple outputs, as opposed to the single output in the representation of Figure 2.6. Yet, songs are ultimately listened to as [9] one single *mixed* signal. Accordingly, a final step is required in DAWs to merge the outputs of the various sound-source-and-effects chains into the required, limited number of signal. This mix-down process (mixing $N > 1$ signals down into a single signal) is often done in a *mixer* unit, which receives multiple inputs (called 'tracks' in the context of mixers) and outputs a limited number of signals, ready for distribution to end users. As part of this thesis, I focus only on

9 Note that this single signal potentially including multiple distinct *channels* for sound spatialization, typically two in the context of stereo music.

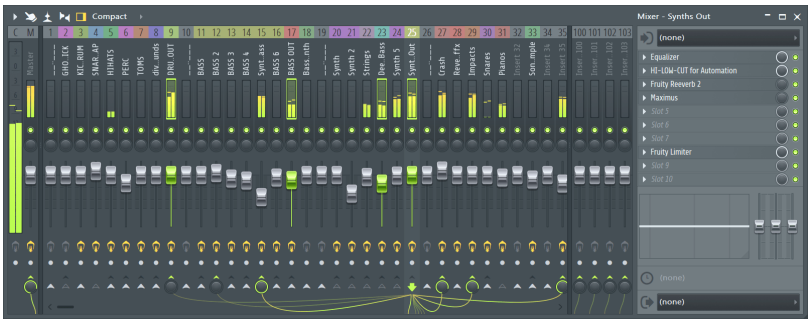


Figure 2.7: A screenshot of the mixer UI element in consumer DAW FL Studio, with the master (stereo) track on the left and individual tracks on the right, each with individual amplitude, stereo panning and effect sections.

controllers and individual sound sources. In particular, I did not work on the design of effect units, and can therefore describe all relevant metaphors in the context of the idealized, simplified representation of Figure 2.6. Nevertheless, the tools developed as part of this thesis are designed and developed with integration with other tools in mind and are ready for usage alongside other tools in real-world music production situations.

I now describe more precisely the computer representation typically[10] underlying these symbolic representations of music, a format which is used throughout the prototypes developed as part of this PhD project in order to allow them to communicate with professional music production tools: the MIDI format.

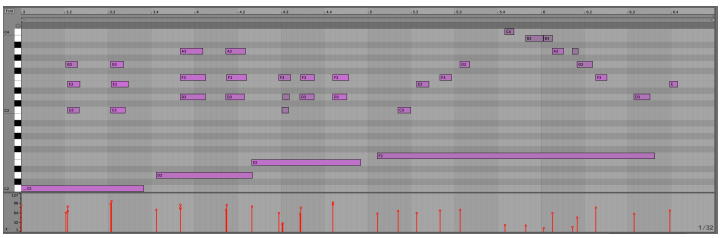
GENERIC CONTROLLERS AND THE MIDI FORMAT In order for digital sound synthesis and processing units to be interoperable, the MIDI standard, a technology for the definition and transmission of controller signals, has been set up in 1983 by a consortium of influential synthesizer makers. At its core, MIDI is an events-based protocol derived from a representation of note sequences inspired by sheet music and piano playing. It consists in a temporal succession of individual *notes* being *played* and *released*. More precisely, the two central events are the note-triggering `NoteOn` and note-releasing `NoteOff` events, which each take as parameters: the exact timing at which the event occurs, the specific note (or *pitch*) being played and, in the case of the `NoteOn` events, the intensity (or *velocity*) at which the note is being activated. One can think of `NoteOn` events as key-presses on a piano with the parameters describing the specific note being played and the intensity

10 Other standards exist such as Open Sound Control (OSC) [42, 116] but these remain more niche and have received less adoption in mainstream consumer DAWs.

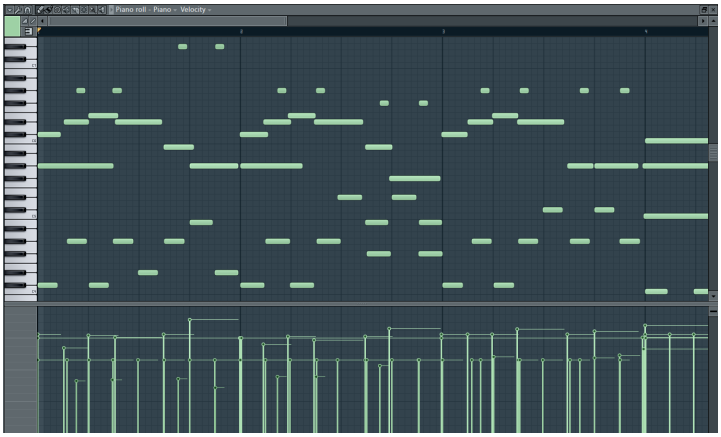
of the physical act of pressing, and `NoteOff` events as describing the release of said notes. According to the previously outlined separation of concerns, controllers emit MIDI commands via a MIDI-Out output (either physical or virtual) and sound sources receive MIDI commands via a MIDI-In input (again, either physical or virtual).

HARDWARE MIDI CONTROLLERS... Typical hardware based controllers include *MIDI keyboards*, which are akin to the acoustic piano, but only send out abstract MIDI commands on each key-press rather than producing sound on their own, or *sequencers*, which are programmable machines that can record and play back MIDI patterns. A MIDI keyboard can be plugged directly (via its MIDI-Out output) into a synthesizer's MIDI-In input to play it. But keyboards can also be used to directly record, at a symbolic level, MIDI sequences into a sequencer! Indeed, programming hardware sequencers by hand can be tedious for complex sequences, requiring many button presses to select the desired time position in the sequence, then cycle through the available notes to set the desired one, repeating this for each note definition. Often, patterns are therefore actually recorded into a sequencer by playing them through an external MIDI keyboard feeding into the sequencer – but this naturally requires the ability to play the piano! A hardware solution has been the development of *step-sequencers*, which sidestep the difficulty posed by individual access to arbitrary note positions by coarsely *quantizing* the available time-scale, forcing it into a *discrete* musical time with limited resolution. With common step-sequencers offering only 4 steps in each single beat, these machines represent an example of devising a trade-off between expressivity and ease-of-use.

... AND THE DIGITAL PIANO-ROLL Similarly, another very widely used UI is the piano-roll, two examples of which are shown on Figure 2.8. The piano-roll is by far the most used representation representation for interacting with *musical notes* in DAWs, if not the only, when one focuses only on the major DAWs such as Ableton Live, Apple's GarageBand and Logic Pro, AVID Pro Tools and Image-Line FL Studio.... Piano-rolls depict a 2-dimensional view where the horizontal axis represents the time and the vertical axis represents musical *pitch*, discretized into successive notes, typically representing the 12 tones of the chromatic scale, C, C♯, D, E♭..., B, over multiple, increasing octaves.



(a) The piano-roll in Ableton Live



(b) The piano-roll in Image-Line FL Studio

Figure 2.8: The piano-roll, represented here in two different flavors, enables to visually represent, arrange and edit sequences of musical notes. Blocks represent individual notes, positioned on a time-and-pitch 2-D grid. Additional note-level parameters such as velocities can be edited through the pane below.

The Piano-Roll, though convenient as a representation, is arguably challenging to use. Indeed, proper musical construction with it technically requires an implicit knowledge of underlying rules of musical melody and harmony, not at all made explicit in the interface. Furthermore, although it technically provides an unquantized representation of time, with the ability to position note events arbitrarily on the time axis, it only provides limited visual cues as to the musicality of these micro-timings. This renders manual, unquantized operation of piano-rolls difficult. As such, they are commonly used in a strictly quantized fashion, as discussed in the essay by Faber [40] mentioned in Chapter 1. Finally, piano-rolls inherently rely on the assumption that the sole notes of a *piano* suffice as a generic representation of music. This is however by no means the case in all generality when moving away from Western musical practices, as discussed above, and these interfaces, although ubiquitous in existing DAWs make it very difficult if not impossible to write music in non-Western scales such as those of classical music of Southeast Asia. These very limitations have actually prompted recent works – namely by musician and researcher Lamtharn "Hanoi" Hantrakul *aka* Yaboi Hanoi – on the application of AI to music creation, which I will discuss in the next chapter on existing approaches to AI music creation.

INTERMEDIATE DISCUSSION This basic review of musical controllers has highlighted the constraining and potentially problematic simplifications these interfaces rely on to make the manipulation of musical events convenient. Moving to the next step in the generic music creation pipeline described in Figure 2.6, we now enter the audio domain and I present the two most common interfaces for sound generation, focusing in particular on the typical interaction scales these two modalities entail.

SOUND SOURCES Two major types of tools exist for the manipulation of sounds. First are *samplers*, devices which play back and transform existing, recorded sounds. These can be seen as operating at a very micro scale, as small as that of individual digital audio samples, reaching sub-millisecond orders. A second widely used modality for sound creation is the use of *synthesizers*, which generate new tones from scratch according to electronic, numerical or physical models. Synthesizers commonly shift away from directly representing sounds in the temporal format, manipulating instead an external, high-level

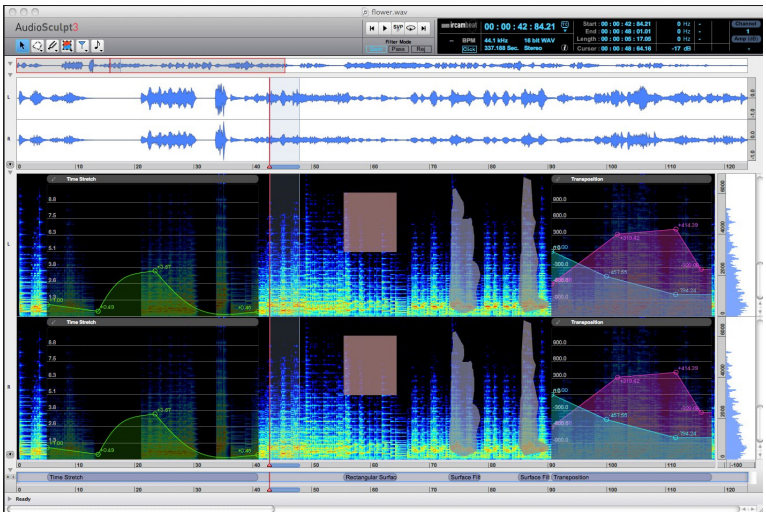


Figure 2.9: A screenshot of the Audiosculpt interface for audio sample transformation.

model of the generated sound, therefore blurring the actual scale of their operations.

SAMPLERS AND SOUND-EDITORS By relying on previously recorded sounds, samplers enable the manipulation of musical object at a very small scale: recorded sounds can be sliced at the scale of individual numerical samples. This enables a vast number of sound transformations. Tools of the Audiosculpt family from Institut de Recherche et de Coordination Acoustique-Musique (IRCAM) are an example of processing sounds in this fashion. I display on Figure 2.9 a screenshot of the (now retired) version 3 of Audiosculpt. This specific interface has now been subsumed by the AudioSculpt As Plugins (ASAP) suite of plugin, which I discuss in a following paragraph on the value of modular, small-scale design of music creation tools, as discussed also by their developer. Nevertheless, the general principles these interface operate upon remain very similar. Practically, these sample-based tools make it possible to visually extract, duplicate, remove time-frequency portions of a sound, or even hybridize multiple existing sounds in the time-frequency representation. Sample editors however suffer from two key limitations. First, the precise, sample-level manipulations they rely on require very precise input capabilities, and are difficult to precisely perform on touch-based devices. This can be seen from the high density of the representations displayed on Figure 2.9. Indeed, even tiny mistakes in re-aligning slices of audio when performing for instance cut-and-paste editing of an audio recording can easily result in audible

clicks. This calls for very careful positioning of the manipulated objects, only achievable on large, precise devices. Second, sample-based tools such as Audiosculpt crucially rely on, well, samples, and these are unable to propose *novel* audio material, let alone novel material coherent with the audio samples already imported within the interface. As such, sample-edition tools such as Audiosculpt solely operate in a, so-to-say, *internal* fashion: all of the content the user can create with them originates solely from the imported samples. Given, this is not a critical issue, since users can always import and *layer* additional samples themselves within the interface, but this certainly makes for lengthy processes, where each novel texture has to be sourced, sliced, extracted, inserted...

SOFT-SYNTHS On the other side of the spectrum are synthesizers, which propose to shift away from the direct representation and manipulation of sound, and mediate its manipulation through a controllable *model* of it. Such models include physical models, like the Karplus-Strong model for string sound syntheis as well as mathematical ones, such as FM synthesis, pioneered by Chowning [20]. These models typically consist of a fixed number of controllable numeric parameters, which modify the texture of the generated sound. As far as interfaces as concerns, typical hardware and software synthesizers (or *soft-synths*), are commonly represented as a collection of rotating-knob-like controls, which can be used to individually set the various parameters of the sound. Sound synthesis programs often represent complex, highly non-linear functions and modern software synthesizers such as the famous Serum (widely used in electronic dance music and movie soundtracks production), displayed on Figure 2.10, sometimes exhibit more than a hundred such individual parameters. Controlling these complex interfaces is already arguably challenging on desktop computers, even with the ability to have most controls on sight. But these manipulations become difficult on smaller devices, where it is commonly impossible to have a global view of the current setup, hindering exploration and control.

2.2.3 A case for mobile music making

This short review of standard UIs for music production highlights the high precision most of these require for fine-grained control, making them are to implement on small, touch-based devices. Yet there

Figure 2.10: Xfer's wavetable-based virtual synthesizer Serum exhibits over a hundred independent controls spanning several tabs

is both a large offering and demand for mobile music production app, as discussed for example in a paper by Koszolko [68], which analyses both subjectively, through the author's mobile music-making experience, as well as through questionnaires and interviews with fellow mobile musicians, the specificities of (iOS-based) mobile music production. The key here, as discussed in the paper, is that these mobile devices do indeed already provide an appealing, hands-on experience, for instance in terms of live music performing. Koszolko [68] cites the following two aspects: "Firstly, tablets and phones offer access to instruments within a lighter, smaller and therefore more portable device than hardware synthesisers or desktop computers. Secondly, the user interaction is *directly connected to instruments' GUI* through hand gestures rather than through additional devices such as MIDI controllers and computer mice." (emphasis mine). That is, users value the ability to operate directly on the visual representation of virtual synthesizer knobs, rather than having this interaction be indirect through e.g. a mouse. Nevertheless, this paper mostly focuses on live music performance, and even though it mentions several practicioners who produce music solely with their mobile phones, little mention is made of fine-grained compositional approaches, instead focusing on:

- direct *audio* recording and sequencing,
- MIDI performance recording with no mention of subsequent note-level edits,

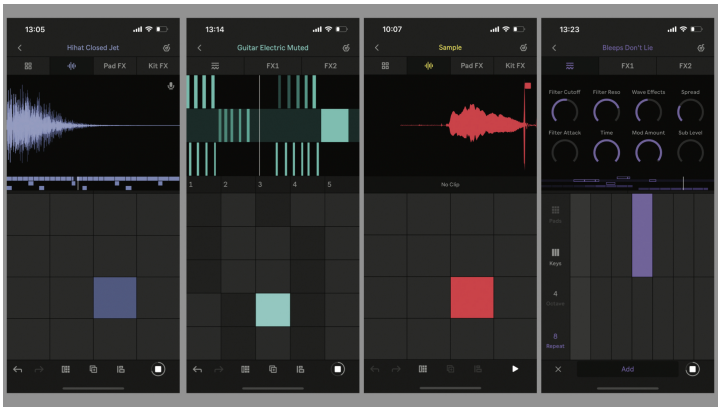


Figure 2.11: Overview of several integrated UIs in mobile DAW Ableton Note for playing and recording performances with percussion samples, tonal samples or keyboard-based synthesizers.

- loosely controled, stand-alone generative models for musical notes.

All of these practices either require users to have instrument-playing skills, often additionally requiring external hardware or to discard low-level control, off-loading melodic and compositional aspects to largely autonomous generative processes.

The same focus on playful performance over fine-grained composition is also to be found in Ableton Note, the recently released and very well received mobile version of influential desktop DAW Ableton Live. As depicted on Figure 2.11, Ableton Note provides several UIs for jamming-like performance, enabling the recording of musical ideas directly on the mobile device's touch-screen and without the need for external hardware. The mobile DAW additionally enables sequencing and triggering these recorded clips on-the-fly for live arrangement.

Ableton Note nevertheless – as of February 2023 – does not appear to offer any form of subsequent micro-level manual editing of these recorded performances, meaning that the only way to fix a mistake in a recorded sequence is to re-record it completely. This usage therefore requires users to be able to properly jam and improvise on the provided interface, potentially limiting the accessibility of the tool. Furthermore, it makes it difficult even for experienced and expert users to perform fine-grained corrections in a goal-oriented composition setting. Based on available reviews on the Apple Store[11], this ability to manually edit recorded sequences instead of only being able to input notes in a jam-like fashion nevertheless appears to be one of the most oft-requested features of the app. Another frequently requested feature would be

11 https://apps.apple.com/us/app/ableton-note/id1633243177

the ability to perform manual, long-term arrangement, which would complement the already available possibility of manually triggering the recorded sequences to combine them live.

For instance, an Apple App Store user review from October, 19th 2022 by user *Wafflesarecrunchy123@* reads:

> Good app idea and great UI, still needs features.
> *[...] I find that every DAW on the market has some sort of piano roll to draw in notes on a clip. Although Ableton's idea for the app might've been to perform everything into the tracks, if someone were to be using AirPods for example, the latency would just make the process a lot more challenging than it needs to be. [...]*

Similarly, user *Ferrichrome* wrote on October 18th, 2022:

> Has potential, missing some features
> *[...] This is a great slimmed down version of session view with choice Ableton instruments and effects included. [...]*
> *However there are some features that are missing that make creation a lot more difficult than it should be. First of all, I'd like to be able to adjust the length of notes in clips.*
> *Also it's weird that you can't create a note manually, you have to play everything in which gets really annoying.*

Finally, in a more recent review from January 21st, 2023, user *That one idiot in the back* says:

> Sequencer setting update
> *[...] Plopping and killing notes with a press of a button would be a million times better than having to only play it live dozens of times to try and get it right. I do love to play how its supposed to when I'm at home and can hear it in real time, but that delay when I'm at work or something really kills the motivation. Having some sort of step sequencing option would be absolutely amazing [...]*

Please note that this selection of reviews is by no means an attempt at discarding the quality of this app. Indeed, this lack of a piano-roll interface could very likely be directly attributed to the aforementioned sheer difficulty of offering convincing solutions for these precise operations on small touchscreens. The aptly-named Ableton Note application therefore – at least for now – focuses on providing a great experience in a *note*-taking context, that is, recording quick sketches, to be subsequently developed into more complex full-scale songs within Ableton Live. This vision of a usage in tandem is made explicit by the strong focus on interoperability provided by Ableton for these tools: indeed, Ableton introduced Ableton Cloud, a Cloud-based storage service, along with the release of Ableton Note. Ableton Cloud enables to transfer projects from Ableton Note to Ableton Live. This

transfer is one-way-only, and Ableton Note sets imported into Ableton Live cannot be re-imported into Ableton Note, again hinting at the (assumed) function of Ableton Note as a scratch-pad for *initial* ideas.

As discussed previously, the assumption that users will have the required hardware as well as the necessary software licenses at hand for developing these initial sketches on a desktop computer does not always hold. This vision ultimately restricts the audience of this tool to somewhat affluent users, with existing mastery of a full-fledged DAW, since they will eventually have to resort to one to perform the work of song arrangement and low-level, precise compositional edits.

With this PhD, I set out to explore if this necessarily has to be the case, and if one could go further and "raise the ceiling" on the level of control offered by mobile devices, whilst maintaining the same level of playful, hands-on interaction these devices are lauded for.

2.3 CONCLUSION AND DIRECTION

This chapter has presented various aspects of modern practices of music creation, focusing on several aspects. First, I tried to make clear how music creation is an inherently difficult task, involving the manipulation of inherently complex objects, blending many different scales of representation. Second, I highlighted how Western music theory, which has evolved over millennia, can be seen as an enterprise in constructing abstractions enabling *effective reasoning over these complex objects*, essentially defining and fixing *specific* scales of representation. Third, I have discussed how these scales of representation (pitches, notes, melodies, harmony, arrangement, movements) – and the assumptions they entail – are directly reflected in existing tools for music creation.

Then, focusing on the resulting scales for interaction, I showed how these conceptual representation scales, although valuable as *visual aids* and abstract *references* for memorization and transmission of music, are actually far from perfect when it comes to interaction and creation, especially in the context of music creation on electronic devices, vastly diffused these days, and this for several reasons.

First, these electronic devices practically consider those abstractions as *exact* representations of the music, as opposed to human performers who rightfully only take those abstractions for what they truly are and what they have been designed for in the first place – abstractions,

references, which not only allow but actually invite to interpretation and deviation. A kick drum positioned of the first beat of a measure will *always* be played *exactly* at the *same* moment on each repetition of this measure by an electronic sequencer. Therefore, in order to produce music with the same expressivity as that reached by human performers, musicians effectively have to work against the various simplifications offered by these representations. This is problematic since it highlights a potentially disproportionate focus on these representations. This is the case for instance of the quantization of the time-scale introduced by rhythmic notations: although they enable convenient and compact music notation for performing musicians, the resulting conceptual quantized grid is pervasive in music creation interfaces such as piano-rolls, which are essentially designed *around* the convenience this quantization enables and can hardly be used as effectively when trying to create music with slight diversions from this grid. Yet, *even with* these coarse abstractions, these tools *remain* complex to use and require interactions with a high degree of precision, that is with a very small scale of interaction. This is highlighted by the discussion I proposed on the difficulties in transferring these different interfaces to modern mobiles devices. I argued that this is a potential issue in light due of the more and more widespread diffusion of these devices as opposed to desktop computers. If anything, it could make for missed opportunities in providing people with proper tools for artistic expression directly *on the devices they have at hand*, a key aspect in facilitating access to music creation to newcomers. This is made even more problematic when one shifts away from considering the Western world only: not only is the imbalance of access to desktop computers compared to mobile devices even more true of non-Western and in particular developing countries, making the difficulties in properly adapting music creation tools to mobile devices even more critical in those regions of the world, the assumptions underlying these tools are actually in direct contradiction with many non-Western musical practices. I accordingly contextualized those criticisms in the more general discourse on the decolonization of music production tools, wherein I have suggested, following works in this field, that this reliance on strong, culturally-constraining assumptions rooted in Western is a critical issue, since it makes creation of music of non-Western cultures challenging, starkly hampering the inclusivity of music creation software.

2.3.1 Problem statement and proposed approach

Scale appears as a crucial aspect of all the presented metaphors for music creation, and I posit that breaking free from the aforementioned limitations could come from directly addressing those scales, in order to propose novel scales, and novel ways of creating music. These novel scales should permit representation at fine-grained scales, enabling the creation of expressive music with e. g. micro-timings, but they should offer effective, higher-level means of interaction, moving away from directly manipulating mote events individually, since this requires a level of precision not conveniently enabled by electronic devices, especially mobile ones.

With the inherent complexity of musical objects in mind, irrespectively of their being perceived or not under the prism of Western music theory, I therefore suggest that in order to go beyond existing scales of representation, radical new approaches should be envisioned, and that statistical modeling, and in particular *deep-learning-based representation learning and generative modeling*, represent very viable opportunities. Indeed, these statistical approaches focus *directly* on *learning higher-level representations of data*, straight from the *observation of the data itself*. As such, these models could hopefully be made to learn valuable, intermediate representations of music at novel scales, enabling more effective interactive composition processes, whilst also being able to capture the fine-grained details and cultural specificities of the music shown to the model during training, ensuring expressivity and cultural relevance of the created material.

Concretely, the goal we set ourselves is to conceive novel metaphors for of creating music that are flexible and rich enough to be considered as viable alternatives to existing standards present in common DAWs such as note-by-note editing on a piano-roll, yet should be significantly more usable. Here, I consider usability both in terms of those novel interfaces being readily appropriable by non-musically literate users, and in terms of these interfaces physically lending themselves to usage on modern, touch-based mobile devices. The notions of flexibility and richness being intrinsically vague, I suggest evaluating this specific aspect through extensive usage of the resulting tools by professional musicians, since these expert users are arguably very best judges for the potential of novel techniques to be viable future alternatives to current music production standards. Concerning the usability, and as far as the work presented in this thesis is concerned, I designed the three interfaces constituting this work with the core constraint that

they should behave exactly the same either on mobile or on desktop, providing the same creative capabilities, thus ensuring that the mobile counterpart would not just be a "diet" version of the desktop version. I additionally evaluated the reception of these techniques by the general public during demo sessions as part of multiple workshops and events, and in particular through the stand-alone exhibition of the NONOTO interface in a museum of computer science in Germany, where attendees could create music on their own with the tool, as described in Chapter 4.

I am arguably far however from being the first to think about applying statistical modeling to the creation of music, and accordingly, I present in the next chapter existing approaches that have equally proposed to train statistical models on musical data for music generation. Nevertheless, I posit that by not specifically focusing on the aspect of scale and rather focusing mainly on advances ion modeling capabilities and quality, existing propositions are regrettably lacking in the interactive modalities they enable.

3 ON AI-ASSISTED MUSIC CREATION

Guided with the overview of music theory and of music creation interfaces, which has prompted me to suggest using statistical modeling as a means of devising novel, more usable scales of interactivity for music creation, I propose a brief literature review of works applying AI to the creation of music. To this end, I gradually build up, focusing specifically on the ability that these approaches offer for controllable music creation, with the assumtion that if these tools are to be seen as legitimate alternatives to existing music creation interfaces, they should provide similar levels of control. In particular, I envision control in the context of DAW-based music creation, accordingly focusing on the ability to perform open-ended creation with successive refinement, along with the ability to revise, modify, expand and generally manipulate the created material in off-line fashion, with random-access to any portion of the work at any moment, as enabled by DAWs.

In the following I call *model* any mathematical object which can be *trained* for various tasks on some corpus of example *data points*. In the case that interests us, the task will generally be that of *generative modeling*, that is, with the example data points being e. g. musical sequences of a given repertoire, trying to use the model to generate novel musical sequences exhibiting similar structure and style as the examples present in the training corpus.

In this perspective, I immediately discard existing AI-driven *"automatic"* music creation systems such as AIVA, Amper, OrbComposer or the now retired JukeDeck. Indeed, the very promise of those tools is to provide single-click music generation, absolutely removing access to the low-level musical representations and only providing very global-scale controls such as dropdown-selectors enabling the choice of a genre and mood for the song to generate. As such, they move the interactivity away from the music itself, arguably reducing the user's grip on the musical material. In particular, they do not allow for any compositional approaches over the generated material, which can either be used as it is or simply discarded, requesting the model to generate another example.

3.1 SHALLOW MODELS

On the other side of the spectrum are tools with limited prior knowledge of music and that are trained *on-the-fly* on examples played by a user, commonly named Interactive Machine Learning (IML).

3.1.1 Interactive Machine Learning and the Wekinator

The seminal work by Rebecca Fiebrink with the Wekinator is representative of the very rich interaction modalities enabled by this framework. The Wekinator initially arose in 2009, in a time when consumer laptop computers were beginning to embed more and more sensors supporting more and more modalities such as webcams for image sensing, gyroscopic sensors or accelerometers. Simultaneously, the video game industry had also given birth to a myriad of cheap external sensors such as Microsoft's Kinect for human pose estimation and depth perception or Nintendo's Wiimote for infrared and motion perception. The Wekinator was proposed as a sensor-agnostic system to build AI-assisted mappings to compatible targets, on-the-fly, by example. Typical output targets for the Wekinator include synthesizers, but the software more generally allows building real-time mappings to any OSC-compatible system. The mapping design is done by associating desired configurations of the target system, to configurations of the input sensor. These input configurations are then recorded and labeled according to the desired target configuration. A lightweight ML model such as k-Nearest Neighbors or small Neural Network (NN)s is then trained on-the-fly on this small, user-defined dataset. Along with interpolation between the different target classes, this thus enables control of the target system via the input sensor, with user-defined mapping. Retraining and fine-tuning of the mapping can also be performed again later on by adding more examples to the existing classes to refine the model's predictions or by adding new classes to enable reaching new configurations of the target system. This very effective framework enables quick yet bottomless prototyping.

Thanks to its ease of use, IML has been adopted and appropriated by many artists, yet it suffers from some limitations. First, it separates the teaching phase from the performance phase by construction, potentially hindering serendipity or spontaneousness in the system. Second, and more directly related to the work performed in the present PhD, the IML approach only works well for very small models that support

on-the-fly re-training in a short time. This therefore forbids usage of large, expert models able to learn deep musical structure such as the ones that have emerged in the music ML community in the last years.

3.1.2 On co-improvisation approaches

Outside of the IML these same considerations for the limitations in using very small statistical models in music creation also apply in the case of other small-scale models of the general Factor Oracles family [4]. Indeed, since the models in those approaches do not bring with themselves any knowledge of musical structure, they can only infer and copy musical structures and patterns as played by their users. Accordingly, novice users are rarely able to play music susceptible of driving the models into generating anything else than the unstructured music they can play. Such tools include models for co-improvisation of the OMAX family, which are able to respond to improvisations performed by their user. In his PhD thesis entitled "Designing With Machine Learning for Interactive Music Dispositifs," notes that computer music systems often rely on complex underlying mathematical models that users should understand to appropriately handle, such as in the OMAX family of tools, where understanding of the underlying Factor Oracle model is key to getting optimal results. This is the case also of musical programming environments such as Max/MSP or Pure Data, which require their users to display substantial programming skills. Additionally, improvisation-driven models of e.g. the OMAX family require users to input some initial musical material, which would again be blocking for novice users, as Scurto writes [97, p. 45]: "Crucially, [such models] still coerce humans into demonstrating expert musical examples to explore new behaviors, which might prevent non-musicians from interacting with these systems."

Therefore, in order to enable even novice users to obtain satisfactory interactions with a AI assisted music creation tool, the underlying should be powerful enough to learn, ahead of time, typical structures of music, so that the users can then, during usage, rely on the model to make musically meaningful proposals. For this reason, we turn to neural modeling, the recent advances of which hold in themselves significant promises for the ability to build such musically-literate co-creative assistants.

3.2 DEEP GENERATIVE MODELING AND INTERACTIVITY

Although an extended review of the wide taxonomy of neural network architectures and related optimization techniques in the context of generative modeling is outside the scope of this PhD, I nevertheless provide a brief introduction to the core concepts underlying neural modeling to help the unfamiliar reader better approach the work presented in this manuscript.

3.2.1 Generative modeling

Given data of interest $\mathcal{X}$ – typically a subset of a vector-space $\mathbb{R}^n$, for instance the subspace of "realistic" trumpet sounds in the general space of signals – one might be interested in being able to sample data from this space, e. g. create new trumpet-like sounds, or transform existing trumpet sounds to introduce new characteristics. Unfortunately, it is usually hard to do so: intuitively, in the case of trumpet sounds for instance, sampling an e. g. ten seconds long signal at random from typical distributions is highly unlikely to generate a convincing trumpet sound (instead, it is more likely to produce unnatural sounding noise). Similarly, for symbolic music, sampling notes at random according to typical, simple probability distributions, *rarely* produces pieces of the same complexity and regularity as "real" music. Generative modeling proposes to tackle this through a two-step approach: first, one defines a *prior* distribution $p(\mathbf{z})$ *which can be efficiently sampled from* and secondly one learns to generate new samples $\mathbf{x}$ from the *conditional* distribution $p(\mathbf{x}|\mathbf{z})$, resulting in the following generative model of $\mathbf{x}$ samples:

$$\begin{cases} \mathbf{z} \sim p(\mathbf{z}) \\ \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \end{cases}$$

In more precise statistical terms, defining a *statistical model* then amounts to defining two families of distributions: first $\mathcal{P}_{\mathbf{z}}$, where each $p(\mathbf{z})$ in $\mathcal{P}_{\mathbf{z}}$ is a candidate prior distribution, and, similarly, $\mathcal{P}_{\mathbf{x}|\mathbf{z}}$ where each $(p(\mathbf{x}|\mathbf{z}))$ defines a potential conditional distribution. One should additionally define a way to *search for appropriate candidates* within those families of distributions, that is find the probability

distributions which best approach the target distributions of interest under an appropriately chosen notion of proximity. This optimization process is usually done by maximizing the *likelihood* of real data under the generative model, provided a dataset $D = (x_1, \dots x_m)$ of independent and identically distributed samples drawn from the true data distribution $p(\mathbf{x})$.

In so-called *variational* inference, one looks for an appropriate prior distribution $p(\mathbf{z})$ within a pre-defined family of probability distributions $\mathcal{P}_{\mathbf{z}}(p_\phi(\mathbf{z}))_{\phi\in\Phi}$, with ϕ the so-called "variational" parameters (usually, Φ is a subset of a smooth space over which one can compute derivatives). Similarly, one looks for the conditional distribution within a family:

$$\mathcal{P}_{\mathbf{x}|\mathbf{z}}(p_\theta(\mathbf{x}|\mathbf{z}))_{\theta\in\Theta}$$

This formulation allows turning the problem of devising these distributions into on *optimization* problem. Indeed, provided a dataset $\mathbf{X}$ of independent and identically distributed samples from drawn from $p(\mathbf{x})$, one can optimize the parameters ϕ and θ by maximizing the probability of the real data under the generative model.

Generative probabilistic models are therefore models $p_\theta(x)$ with variable parameters θ that seek to approximate the "*true*" probability distribution $p(x)$ *and* can be efficiently sampled from, so that after an initial optimization phase to devise appropriate parameters θ, one can sample new data points $x \sim p_\theta(x)$, using the generative process $\mathbf{z} \sim p(\mathbf{z}); \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$. These new samples are expected to follow the original, hard-to-sample-from distribution.

Neural modeling, the specific case of relying on *neural networks* for realizing the variational families of distribution p_θ and p_ϕ, has been proposed as a way to efficiently construct statistical models and has demonstrated significant performances over the last decades.

3.2.2 A primer on neural modeling

Generally speaking, *deep generative modeling* is defined as the use of *multi-layer* neural networks for modeling the distribution of training data (typically represented as multidimensional matrices) in such a way that one can subsequently sample for this approximated distribution and retrieve new data points similar to the training data. Each of the layers in a deep neural network usually consists in the application

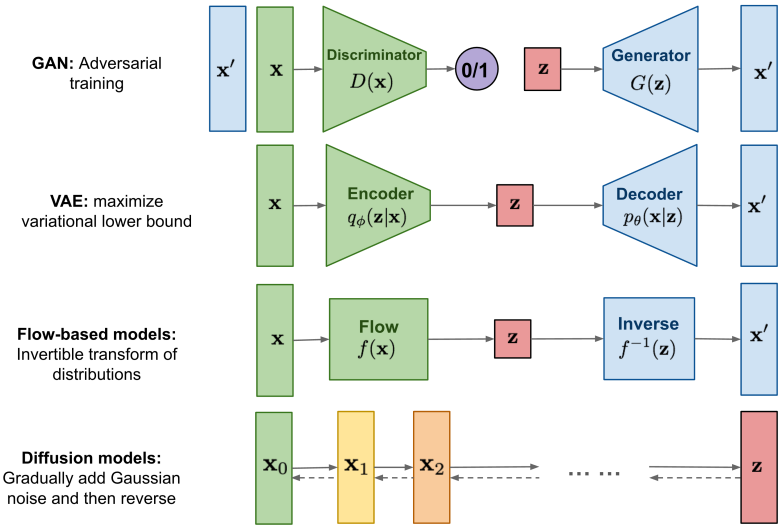


Figure 3.1: Overview of different types of generative models. Picture taken from Weng [113].

of a linear operation on its input followed by the application of a non-linear function, called the *non-linearity*. Typical linear operations include matrix projection with a learnable matrix in so-called *Dense* or *Fully Connected* [1] layers or convolution with a learnable local kernel in Convolutional layers, though there are many more. Here "learnable" means that the parameters or *weights* of the associated matrixes are variable, optimizable parameters of the resulting statistical model.

THE *DEEP* IN DEEP NEURAL NETWORKS The term deep in deep neural modeling refers to the fact that the associated networks are composed of a stacked succession of multiple, elementary neural layers. The underlying assumption (and experimental observation) here is that such models are therefore able to *progressively* – layer after layer, projection after projection – shape the input data distribution and learn a complex, step-by-step projection and transformation of it onto an ultimately smoother (and easier to explore and sample from) distribution, located in a different vector space. This is in essence the so-called *manifold hypothesis*, presented thoroughly in a classical blog post by Olah [83].

On Figure 3.1, graphical models are displayed for most of common structures of neural networks currently used for generative modeling. Although these different frameworks have crucial differences, they all have in common the reliance on an auxiliary variable z which is used

1 Both names appear commonly in the literature.

to drive the two-step generative sampling process $\mathbf{z} \sim p(\mathbf{z}); \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$. Accordingly, control over the resulting generative process for new data examples x directly derives from the ability to control the sampling of the variable z. This crucially hangs upon the *structure* of z, and in the following section, I present how applications of neural modeling to music creation have approached the design of these control features.

I now list some recent works on music generation, both in the symbolic and signal/audio domain. In doing so, I try to highlight their respective benefits, but also their shortcomings in terms of proposing challenging and user-friendly interaction whilst remaining powerful and flexible enough to be of interest to professional musicians.

3.3 PRIOR WORK ON MUSIC AI

3.3.1 Interactive deep audio modeling

Global conditioning labels such as pitch and instrument family set aside – which can be enforced through categorical variables provided as one-hot-encoded input to the networks –, introducing *local* control in deep learning models of audio is made difficult by the typically high sampling frequency of sounds in waveform representation. Indeed, this high sampling frequency makes autoregressive modeling of sounds challenging. This is the case for autoregressive, waveform-based models of the WaveNet family [84, 109], and such models are bound to operate with low receptive fields and display highly unstationary behavior. Controlling these models therefore typically requires very dense conditioning sequences, sampled at a rate close to the audio rate, making these impractical for manual control. Conversely, models such as GANSynth [35], that use spectrogram representations and generate full-scale sounds of several seconds in parallel through transposed convolutions conditioned on a single vector, lose the ability to control the small-scale structure of the sounds.

Promising solutions have been proposed to introduce control at a mid-level range in autoregressive models through the use of Style Tokens and similar techniques [34, 112]. Here, an intermediary, subsampled latent space is learned along with the autoregressive decoder and used to condition the decoder. Nevertheless, the resulting conditioning vectors are real-valued vectors with unidentified dimensions, making them hard to interpret. Existing control schemes are therefore

limited to extracting the sequence of conditioning vectors of an example sound and transferring it to another sound, but further control is difficult.

I finally mention the impressive results in voice conversion – in the context of speech – or tone transfer – in the context of musical audio – tasks. Voice conversion allows, through a disentangling of different aspects of speech, to alter the speaker identity of a given speech sample [8, 69, 88]. Such disentangled approaches were recently brought to musical sound synthesis through the Differentiable Digital Signal Processing (DDSP) architecture [36], which consists in a set of basic differentiable modules for various classic elements of sound synthesis, such as an additive synthesizer or a convolutional reverb. Combined, these allow to design differentiable synthesis architectures with strong inductive biases which can then be trained to imitate a given instrument with very limited data (in the order of 10 minutes of audio). This disentangled architecture then allows to extract e.g. the F0 contour of a target audio sample and feed it to the model for instrument identity-conversion. Since DDSP is very lightweight, this enables exciting *Tone Transfer* applications as shown in several web-demos by the Magenta team at Google, in an effort lead by Researcher Lamtharn "Hanoi" Hantrakul [52, 53]. These demos enable on-device conversion of input singing to different instruments such as violin or trumpet as well as three classical Indian instruments: the Bansuri, the Shehnai, and the Sarangi. A real-time implementation of DDSP has even been released as a VST plugin, for integration into DAWs [123], highlighting the desire to provide these tools to music producers as well. Yet these conversion-based approaches, which operate as black boxes able to translate into a given set of target identities, do not directly translate to open-ended musical sound creation where a target sound is not available for conversion beforehand. Furthermore, these models, by not providing full structure modeling allowing to sample new sounds from scratch, require an initial sample to modify, which, again, is not always desirable for open-ended creation.

Similarly, the Realtime Audio Variational Encoder (RAVE) model by Antoine Caillon [16] is designed for realtime synthesis of high-fidelity musical data. Thanks to a compact representation based on a multi-band decomposition of the input audio signal, RAVE operates in real-time on consumer CPU-based devices. Prior-distributions over the latent representations computed by this Variational Auto-Encoder (VAE)-based model can also be computed, enabling directly

generating long-form sound sequences by exploration of the model's latent space through the prior distribution. Yet, the authors do not offer any mechanism to finely control this sampling mechanism for guided composition, this lending itself very well however to free-from sampling and exploration. Augmenting the RAVE model with an intermediate-scale control modality such as the one driving the NOTONO prototype I describe in Chapter 7 could prove a valuable approach.

3.3.2 Existing interactive models of symbolic music

As far as integration into the practices of music producers is concerned, the case of symbolic music generation is arguably even more challenging. Indeed, even though – as mentioned – these tools crucially require interoperability for integration within musicians' workflows, there is no open standard – as opposed to the availability of the VST standard for neural sound synthesis – for programmatically controlling the content of MIDI tracks in consumer DAWs, calling for the development of custom interfaces with innovative solutions to this limitation. Solutions have included either standalone interfaces [13, 57, 92] or ad-hoc systems limited to one specific host. The latter have included Max4Live-based Ableton Live plugins [49, 92] or an integration into the (commercial) sheet music creation software MuseScore [47]. These integrations can be poorly accessible for research purposes when the host software in question requires users to own a commercial license, as is the case for Ableton Live. Existing standalone interfaces such as the Magenta Studio suite [92] and Calliope [13], target measure-wide generation, thus lacking the ability for short-term generation at the note-level.

Closest to our goals and a close cousin to our own work on NONOTO, presented in Chapter 4, is the Bach Doodle by Huang et al. [57], depicted on Figure 3.2. The Bach Doodle is a large-scale demo that was released by the Magenta team at Google which enables on-device harmonization of a user-input melody via the COCONET model [56]. Whilst the global scale at which this demo was released represents a very impressive and first-of-its-kind achievement, this interface only targets one-shot music generation (the user inputs a melody note by note, then the model generates a harmonization) and does not allow for iteratively refining the results of the network, crucial for adoption of these techniques by expert users.

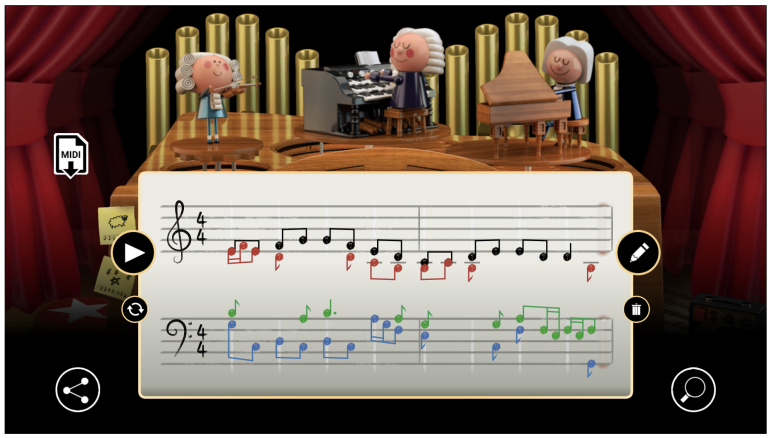


Figure 3.2: The Bach Doodle enables harmonizing a user-input melody (black notes) into a 4-part chorale in the style of J.S. Bach by generating the remaining 3 voices (in red, green and blue). (Picture taken from the blog post announcing the demo [21].)

Conversely, web-based user interfaces such as Yotam Mann's A.I. Duet system [128, 129] and Piano Genie by Chris Donahue [31] offer intuitive, mobile-ready usage but discard offline, long-term composition in favor of playful, real-time improvisation. A.I. Duet operates in a similar fashion to models of the OMAX family [4] and performs on-the-fly sequence completion: the user can play some notes on a piano, then pause, and the model jumps in to continue playing some piano in the style of what the user played via on-device neural network inference. Piano Genie is an Long Short-Term Memory (LSTM)-based model and simple web interface that allows to map a simple 8-button keyboard into based a full 88-key piano in real-time. Users can then improvise on this 8-button controller and the model expands these 8-button live performances back onto 88 keys. This is done via an encoder-decoder LSTM architecture which encodes each incoming note onto one of 8 discrete latent cells, then decodes these cell activations back into 88-keys. The resulting model, depicted on Figure 3.3, is trained on classical piano performances and is ultimately capable of turning sequences over the 8 discrete latent cells (that is, performances on the 8-button controller) back into full-fledged piano performances over 88 keys via the LSTM decoder.

Although technically very relevant and certainly fun to play and improvise with, these interfaces do not offer any solution for long-term planning and composition, potentially limiting their value for musicians. Differing from the purely online, improvisational approach embodied by these two prototypes, the PIANOTO interface I present in

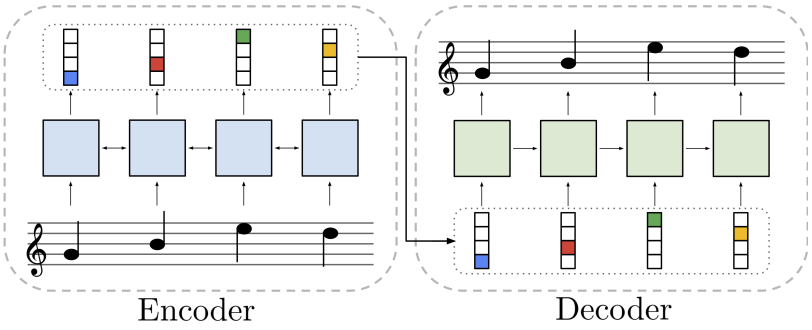


Figure 3.3: The model behind Piano Genie (here shown with a latent space made of 4 discrete cells instead of the 8 cells used in the actual demo). (Image taken from [30].)

Chapter 5 seeks to strike a balance between playful, reactive interaction *and* long-term editing capability for structured music composition.

3.4 INPAINTING

In light of this brief review of existing approaches and since, as discussed in the introduction, it appears that in order to offer convincing interaction mechanisms in end-applications, interactivity should be directly built into the AI architectures developed, I focus in this thesis on AI models designed for a specific task: *inpainting*.

Inpainting models generally deal with the reconstruction of missing or hidden parts of input data. Statistically, this amounts to modeling local, conditional probability distributions. Equipped with such models, it is then possible to "fill in the blanks" by computing the probability distribution for the chosen hidden parts conditioned on the available surrounding context. Inpainting techniques have most broadly been applied to natural images, in which context they amount to modeling the local probability distributions of a pixel given the surrounding pixels. Through this local aspect, we believe that these models offer a promising approach to interactive applications, by allowing users to selectively resample specific parts of a given media, as can be seen on the demo presented on Figure 3.4. Since these models are built with interactivity at their core, they lend themselves very naturally to the development of interfaces. These interfaces should simply enable users to selectively erase parts of an object of interest and let a model reconstruct those, taking in account the remaining surrounding information.

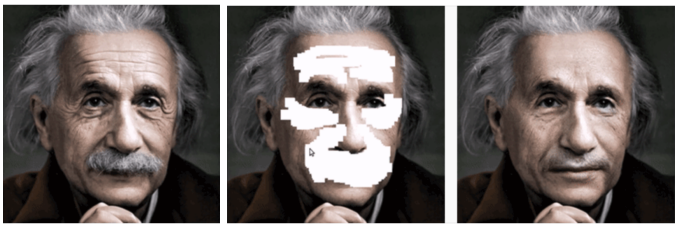


Figure 3.4: An example of image inpainting, taken from Liu et al. [71]. Left: original picture, center: user-input inpainting mask, right: inpainted picture.

The three interfaces presented in this thesis each operate on distinct types of musical objects, but share the reliance on inpainting as the core interactive modality.

3.5 A CASE FOR SMALL-SCALE, MODULAR TOOLS

Finally, in the context of the previous overview of interfaces AI-assisted interfaces for music creation, I provide some arguments for one aspect of the design approach undertaken with this PhD, namely the focus on developing small-scale, single-task-oriented and *modular* tools.

As mentioned in Chapter 1, proper evaluation is a crucial and difficult aspect in both interface design and in statistical modeling. Following the ideas discussed in Wanderley and Mackay [111], I posited in the introduction that valuable evaluation for both the proposed interfaces and for the capacities offered by their underlying models could arise from extended, *longitudinal* studies of the usage their intended end-users make of them, in the actual *ecological* [18] context for which they are designed. In our case, this amounts to trying to observe in particular how musicians react to and subsequently handle the concepts and tools I propose.

Arguably, reaching such ecological validity of evaluation is difficult, since this requires building many functionalities around the core technological innovation, in order for said innovation to be readiliy integrated into the usual workflow of music producers, which is key to ultimately enabling appropriate, *in situ* evaluation. This is an argument for a modular, *plugin*-oriented approach, where the developed artifacts take the form of simple modules with a functionality strictly focused around the novel research approach of interest, but which offer standard means of interoperability with existing consumer applications. This reduces the technological surface and development

cost of these artifacts to simply a. building the core "research-driven" functionality and b. implementing support some appropriate interconnection standards and/or import and export modalities to facilitate integration. This plugins-based approach stands in stark contrast with the development of large-scale tools that integrate too diverse a feature set, making maintenance and further development difficult, whereas each of said features, unless made necessary for the intended design of the tool, could be extracted in a separate, individual plug-in.

An example of this shift from a monolithic software development approach to a modular design in the field of music creation technology is the ongoing re-development of analysis-driven music creation tool Audiosculpt [10, 11]. Developed at IRCAM from the 1990s to the end of the 2010s, this software enables high-quality visual editing of sounds via visual representations. Unfortunately, this tool has been incompatible with modern version of macOS since 2019 and, as discussed by the developper in charge of its undergoing redesign and modernization at IRCAM, Pierre Guillot, in a presentation at IRCAM's Forum, a yearly musicians-oriented symposium, in 2022[2], its monolithic design made it difficult to efficiently update it for compatibility with these new versions of the Apple operating system. Furthermore, the original Audiosculpt had been designed as a fully standalone audio editing tool, with no convenient means of integrating it with external software. This prompted a complete overhaul of the Audiosculpt tool, now designed as a suite of independent plugins and aptly called ASAP, for *AudioSculpt As Plugins*, each re-implementing individual features of the original Audiosculpt and with a strong focus on integrability into existing DAWs, using modern standards such as the Audio Random Access (ARA) technology, enabling a tight, bidirectional integration between the plugins and their host software. As Pierre Guillot states in his presentation, this redesign has helped significantly speed up the development cycle of the Audiosculpt tools, by offloading many features such as mixing or temporal automations to the host software, features for which DAWs are actually primarily designed and for which they have been thoroughly battle-tested. Finally, the added interconnection capabilities now make for even more flexible use cases, at no cost for the users and the developer.

The approach I take in this work, creating three separate and very focused yet interconnectable prototypes, tackling each a specific repre-

2 https://forum.ircam.fr/article/detail/partiels-and-asap/

sentation of music, is ultimately driven by the very same motivations as those driving the development of ASAP.

3.6 CONCLUSION

The previous short review of models and applications for AI-assisted music (co-)creation highlights the difficulty in finding an appropriate balance between ease of use, crucial for novices as well as to attract new users, and the ability for structured, expert usage of these tools. In the following two parts of the manuscript I present, through three AI-driven interfaces, the approach I propose for constructing such interactive approaches.

Part I

INTERACTIVE INPAINTING OF SYMBOLIC MUSIC

General introduction and motivation

In this first part of the PhD manuscript, I describe the design and implementation of two reactive, model-agnostic web interfaces for symbolic music creation by inpainting, tackling two, increasingly complex levels of musical expressivity. These interfaces go from quantized, symbolic music with a rigid time-grid to expressive, un-quantized performances, and as such directly follow the recent advances in symbolic music modeling.

In both of those interfaces, the interaction modality is a simple one: rather than manually selecting and placing notes – a task that requires expert knowledge of music –, the users can select temporal sections of the current musical content of the interface. Upon such a selection, and thanks to context-aware modeling of the past and future notes around this region, the relevant backend AI models generate an inline replacement proposal for this region, coherent with its context. This is a direct realization of the proposal expressed in Chapter 1, wherein we propose to only rely on the fine-grained representation (sheet, piano-roll) for *visualization* but mediate this representation for *interaction* and *creation* through a co-creative assistant operating at a higher, intermediate scale.

The first interface, NONOTO (described in Chapter 4), operates on quantized, fixed-tempo sheet music with neither velocity nor micro-timing information. In particular, NONOTO has been designed to work well with the DEEPBACH model by Hadjeres, Pachet, and Nielsen [47]. Nevertheless, this prototype is essentially model-agnostic and in reality only relies on the sheet representation, as demonstrated by usage with other models of sheet music (Folk songs, Jazz and Pop leadsheets), prompting the inclusion of additional control mechanisms. As for the usage of this prototype in music creation contexts, operating over sheet representations – based on a fixed, quantized time-grid – has a nice side-effect of ensuring that the outputs of this interface are always perfectly stable in tempo and "in time", exhibiting metronomic precision by construction. NONOTO can therefore be seen as a smart, AI-assisted polyphonic *sequencer* mediated by a visually accessible interactive sheet representation and ready to be used in synchronization with other musical sources in real-world music creation settings. To fully realize this potential and ensure that this tool can be beneficial to artists, I integrated support for MIDI-Out and Ableton Link within

this interface, effectively enabling its use in connection with external, professional DAWs, with the ability to independently route each of the voices of the generated piece to different MIDI channels, and instantiate those with different sounds. The flexibility of this integration furthermore opens up interesting aesthetic diversions of the initial training data of this tool, described in the chapter.

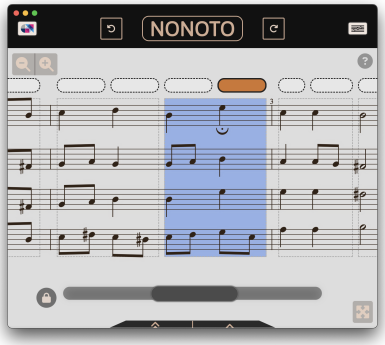
Then, in Chapter 5, I present the second interface for symbolic music creation developed as part of this PhD, PIANOTO. This prototype was designed around the Transformer-based PIA model by Hadjeres and Crestel [49], and helps users edit existing or generate entirely new piano performances with expressive velocity and free-form, unquantized timing, in the form of piano-rolls. As such, it enables the creation of music with significantly more expressivity than what was possible with NONOTO. This radically novel, reactive approach to interacting with piano-rolls, purely through temporal selection in a "point-and-click" fashion and doing without the need to painstakingly position tiny rectangles on an unforgivingly small and precise grid, drives a desire to approach the piano-roll *as an instrument* powered by dynamic "one-to-many" interactions as opposed to the static, expert interface it tends to be. In particular, thanks to the resulting simplicity and lower requirements for precision of this interaction, this interface lends itself well to usage on mobile devices, a significant improvement over existing piano-rolls. I describe the advantages, and current limitations of this approach, in particular, the loss – for the time being – of any form of adherence to an external time-grid, making this tool slightly less directly usable in conjunction with external sources. This however, could very naturally be solved by additional conditioning of the underlying models and is by no means an inherent limitation of the interface's design.

Thanks to the simple interaction modality brought by AI-assisted inpainting, I designed these two interfaces with a mobile-first approach, meaning that they can be used equally well either on desktop or on smartphones, helping bridge the gap between these two platforms when it comes to professional music production. Additionally, since these AI assistants are easy to use and require no little expert musical knowledge, we hope that these concepts could help reach a wider, non musically-literate audience and hopefully spark interest in music composition and music production.

4 | NONOTO

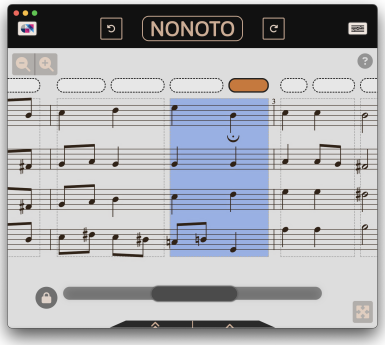
The Well-Tempered Latent

> NONOTO et NOTONO sont dans un bateau [...]
>
> — Dr. Gaëtan Hadjeres

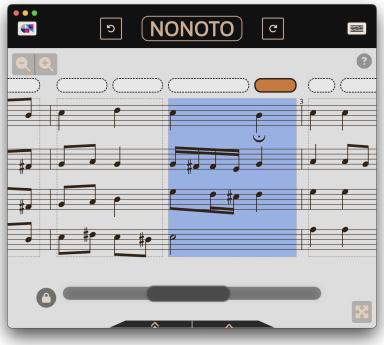


(a) Initial chorale, generated by DeepBach

(b) Click to regenerate



(c) Inpainting result #1



(d) Inpainting result #2

Figure 4.1: Using NONOTO along with the DeepBach model to generate several variations (4.1c and 4.1d) of a sequence through repeated inpainting of the same zone.

NOTE The work discussed in this chapter was initiated as part of a pre-doctoral contract with Sony CSL and constitutes the basis – both conceptually and technically – for the subsequent prototypes developed during the PhD. This chapter is an edited and updated version of a paper that was published at the 2019 International Conference on Computational Creativity [5]. I furthermore presented the NONOTO interface within the Demos track at NeurIPS 2019, held in Montreal, Canada and at the 2019 SONY CSL Open-House in Tokyo, Japan. Development and improvement of NONOTO continued throughout the PhD, with updates to the visual theme (to align it with the other two prototypes developed during the PhD) as well as to the audio engine, bringing in improvements developed as part of building NOTONO and PIANOTO. In particular, setting up a stable and robust synchronization with Ableton through Ableton Link was challenging and took several tries to get to the current (satisfactory, yet still somewhat hack-y) solution, described below in Section 4.4.

Before jumping into the content, feel free to take a look at two short demo videos of NONOTO in conjunction with the DEEPBACH model. The first simply depicts the core interaction mechanism, as shown on Figure 4.1[1]. The second video[2], also created and recorded by myself, shows usage of NONOTO in synchronization with Ableton Live via Ableton Link and with all different voices realized by synthesizers and samplers within Ableton Live, to create a draft of Baroque Acid House.

In this chapter I infrequently switch from *I* to *we*, with my usage of *we* usually depicting Gaëtan Hadjeres and myself, hinting back at the decisions at play when this work was initiated, first at Sony CSL Paris.

INTRODUCTION

In this chapter, I present NONOTO, a cross-platform, web-interface for interactive music generation based on context-aware editing of sheet music. As shown on Figure 4.1, NONOTO presents users with a traditional sheet music, augmented with time-aligned, clickable-overlays, at an intermediate scale ranging from one quarter note to a full measure. By clicking on or touching these overlays, users trigger the underlying statistical model to regenerate the corresponding temporal-slice of the

1 https://youtu.be/jgQVJm59BUw
2 https://youtu.be/iFV0QO59bpA

sheet, conveniently located visually directly beneath it. This modality of interaction enables users, regardless of their knowledge of the underlying rules of melody and harmony, to create music simply by listening.

This work was initially motivated as an attempt to design a web interface for the – then recently released – DeepBach model by Hadjeres, Pachet, and Nielsen [47]. DeepBachis an LSTM-based model which combines two unidirectional temporal models of music, the first going in Although this LSTM-based model is arguably slightly outdated by today's ML standards, the concepts it is based on therein as well as the technologies used as part of building this interface remain very relevant for the subsequent projects, as they directly deal with the design of a high-level interactive representation over sheet music, precise enough to enable focused transformations, yet remaining intuitive enough to enable the generation of convincing polyphonic music by anyone, either on desktops or smartphones.

By focusing on an already high-level, abstract representation of music, these models treat time in a *quantized* fashion (musical, grid-based rhythm) and the generated results therefore exhibit tight tempo and rhythm *by construction*. This arguably limits the expressivity of these models, as they are unable to depart from that fixed-grid and generate micro-timings or deviations from perfect pitch. Yet, this has the advantage of making the outputs of these models readily suitable for playback alongside other sources, since the adherence to (Western) musical rhythm and pitch emsures perfect synchronization to those other sources.

To make use of this, I integrated support for Ableton Link [45], an open, network-based technology for distributed DAW synchronization. Thanks to the integrated real-time MIDI output, NONOTO can be used as a "smart", melodic and harmonic sequencer, in conjunction with e.g. Ableton Live. It is nevertheless also aimed at researchers as it offers a simple and flexible API allowing them to connect their own models with the interface. The interface, released under an open-source GNU General Public License (GNU GPL) license, can furthermore be hacked to integrate dataset specific annotations, which I demonstrate through usage on several models, focusing on several music genres: using fermata annotations in chorales in the style of Bach, and using chord annotations for jazz leadsheets.

Drawing inspiration from then recent advances in interactive interfaces for image restoration and editing [61, 62, 117], we focused

on providing an interface for inpainting models for symbolic music, which are models that are able to recompose *a portion of a score* given all the other portions. The reason is that such models are more suited for an *interactive* use (compared to models generating a full score all at once) as they let users play an active part in the compositional process. As an outcome, users can feel that the composition is the result of their work and not just something created by the machine. Furthermore, we believe that allowing quick exploration of musical ideas in a playful setting can enhance creativity and provide accessibility: the "technical part" of the composition is taken care of by the generative model which allows musicians as well as non experts in music to express themselves more freely.

In particular, the combination of a web-based music creation interface with Ableton Link through end-user-friendly packaging of the web interface as an Electron application seems to be a novel approach, with existing web-based AI-assisted music creation interfaces not providing means to synchronize with external devices.

The code for the interface is distributed under a GNU GPL license and available, along with details for running the DeepBach model as a Docker image[3] and a video demonstration of the tool for electronic music creation in synchronization with Ableton Live, on the `music-inpainting-ts` GitHub repository[4] [119].

4.1 PROPOSED INTERFACE: NONOTO

NONOTO is presented as an interface that displays a musical score along with interactive, overlay boxes over this sheet, as shown on Figure 4.1. Users can modify the score by regenerating any region simply by clicking/touching it. The displayed musical score is updated on-the-fly, without interrupting the music playback.

The generated scores can be seamlessly integrated in a DAW using the Web MIDI and Ableton Link support so that the user (or even other users) can shape the sounds, add effects, play the drums or create live mixes. This creates a new jam-like experience in which the user controlling the A.I. can be seen as just one of the multiple instrument players. Since our approach is flexible, our tool can also be used in conjunction with other A.I.-assisted musical tools like the

3 A matter of a single command, provided that one has already installed the Docker runtime on their machine, and a straightforward installation process otherwise.

4 https://github.com/SonyCSLParis/music-inpainting-ts

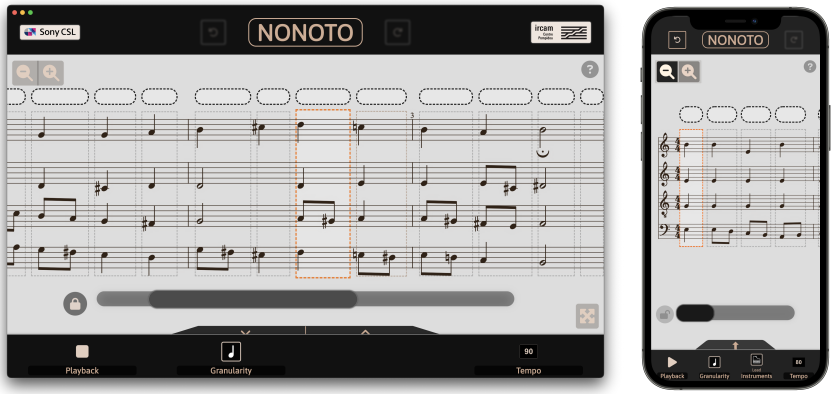


Figure 4.2: The NONOTO interface, used with the DEEPBACH model, in both desktop and mobile display. Additional boxes above the sheet enable the setting of fermatas to guide the model and structure the generated chorale.

aforementioned plugins from Magenta Studio. This interface thus has the potential to stimulate a new environment for collaborative music production and performance.

In the following, I focus mostly on the pairing of NONOTO with the DEEPBACH model, which I denote as NONOTO-DEEPBACH. Indeed, I believe it captures essential use-cases NONOTO, thanks to its polyphonic aspect and to the simultaneous richness and homogeneity of outputs offered by the DEEPBACH model. The ideas nevertheless apply to using NONOTO with other context-aware editing models of (polyphonic or monophonic) sheet music than DEEPBACH and we encourage practitioners to adapt NONOTO to their use cases and connect it to their own models.

4.1.1 Discoverability of the NONOTO interface

The interface integrates in-browser, on-the-fly audio rendering of the music sheet via `Tone.js` and a MIDI rendering of the corresponding MUSICXML sheet representation. Through the use of clearly visible animations synchronized with the audio-playback – namely, a dynamic enlargement of the displayed overlay boxes when they contain the current playback position, I sought to ensure that even users who cannot read music at all could properly locate themselves on the sheet and build a direct, audiovisual mapping to the range of the performance that would be modified by clicking on any given box. These animations, along with an automatic "call-to-action" animation, which triggers the visual blinking of a random subset of the overlays boxes,

where furthermore designed as a way to promote the discoverability of this interface. This goal was in particular directly prompted by an invitation made to my advisor Gaëtan Hadjeres by Dr. Doreen Hartmann to display his DeepBach model at the Heinz Nixdorf MuseumsForum of computer science in Paderborn, Germany as part of their 2019 exhibition "*Mensch, Roboter! Leben mit Künstlicher Intelligenz und Robotik*"[5] on modern, interactive applications of AI and robotics. We proposed to showcase DeepBach through an early version of the then-burgeoning NONOTO interface. With the exhibition taking place in Germany across several months, it was not possible for either Gaëtan or me to be there and explain the intended usage of the tool to visitors. This prompted me to try and ensure that potential users could discover how to initiate interactions with NONOTO without the need for any external guidance. Additionally, this desire for the application to be self-explanatory was a further argument for a somewhat minimalistic design of the interface, trying in particular to remove any unnecessary buttons, since each additional button with an independent, specific functionality makes for an increased cognitive load and a risk of discouraging newcomers. This was furthermore another argument for removing the otherwise somewhat ubiquitous *"Generate"* button of AI-assisted interfaces and make what would have otherwise simply been *selection* boxes directly *interactive themselves*.

The largest computer museum in the world as of 2018.

4.1.2 Adaptability of the NONOTO interface

We consider adaptability of NONOTO across two dimensions: adaptability for developers and adaptability for musicians. Adaptability for developers is obtained through the use of standard, modern web API such as the Web Audio and Web MIDI APIs, making it easy to hack the interface for new types of data and models, integrating support for specific control modialities. Adapatability for musicians, meaning the capacity for musicians to take NONOTO and easily integrate it into their typical workflow is made possible via the support for standard interconnection and synchronization modalities: MIDI and Ableton Link, enabling usage alongside consumer DAWs. I now present these two dimensions with more details.

ADAPTABILITY OF NONOTO FOR DEVELOPERS AND RESEARCHERS

Hacking the open-source NONOTO interface with the use of modern

5 *Human, robot! Living with Artificial Intelligence and Robotic*

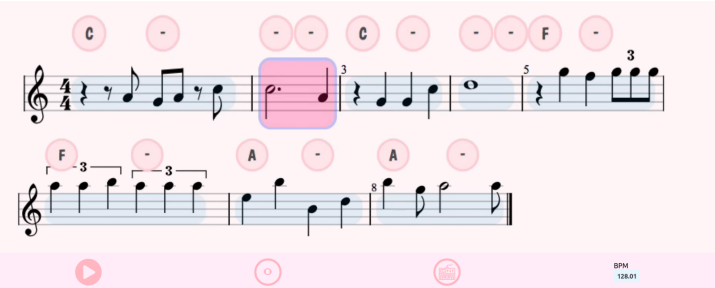


Figure 4.3: An early, *Marie-Antoinette-styled* version of the NONOTO interface showcasing usage with a model of music trained on a dataset of leadsheets, with support for chord annotation constraints.

web technologies makes it possible to integrate additional control modalities, that is, other means of controlling the models via auxiliary conditioning, depending on the domain-specificities of the models used. These additional forms of conditioning are sent over the network, along with the current state of the sheet, to the generative model when performing re-generations.

Presently, NONOTO provides for instance the ability to position *fermatas* when performing Bach chorales generation, as shown on Figure 4.2. Fermatas operate as musical punctuation, helping introduce structure in music by means of intermediate sequence resolutions and pauses and are supported as additional conditioning by the DEEPBACH model. Similarly, when used with models of jazz lead-sheets or folk songs such as the generic ANTICIPATIONRNN model these features where initially built upon [50], NONOTO supports the setting of *chord progressions*, as shown of Figure 4.3. The selected chord symbols, for each portion of the sheet, are sent to the back-end models when performing regenerations, thus making it possible, with appropriate models supporting this type of conditioning, to easily generate melodies whilst following chord progression constraints. Support for this additional control modality was enabled by making the chord annotations interactive as well.

ADAPTABILITY OF NONOTO FOR MUSICIANS: DAW INTEGRATION In order for NONOTO to be readily usable in traditional music production and performance contexts and fully realize its potential as a "smart", AI-assisted sequencer, I implemented the possibility of integrating the generated scores in any DAW in real time. To this end, I provide the user with the option of either rendering the generated sheet to audio in real-time from within the web interface using `Tone.js` (for standalone usage) or of routing it via MIDI to other musical sources such as

synthesizers or sampler running within a DAW on the user's machine. This integration was made convenient thanks to WebMidi.js[6] [121, 25], an easy-to-use abstraction over the Web MIDI API. This makes it easy to generate sequences from within NONOTO but have them rendered by an external synthesizer. When generating polyphonic sheets or when working with e.g. lead-sheets providing both melodies and chord progressions (rendered as a separate voice), each of the individual voices is emitted over a *different* MIDI channel, allowing users to painlessly have each of those voices rendered by different instruments. I additionally integrated support for Ableton Link[7], an open-source technology developed by Ableton for easy synchronization of musical hosts on a local network, allowing to synchronize the interface with external DAWs such as Ableton Live (on macOS and Windows) or Bitwig (on Linux, macOS and Windows). I present the technical side of this integration in Section 4.4.

Adding support for these technologies does not represent novel advances on our side *per se*, yet, paired with the support of arbitrary generation back-ends, they allow to quickly test new generation models in a standard music production environment with minimal overwork and make for a beneficial tool for researchers – and the first of its kind to our knowledge.

These flexible integrations open up the way to interesting aesthetic diversions. Indeed, NONOTO – for instance with DEEPBACH – can ultimately be conceived of as generating independent yet *harmonically cohesive* MIDI signals, thanks to the underlying generative model having access to and control over each of the 4 generated voices at each step. These four voices can then be used to control and prime multiple elements of a song, without the requirement to strictly adhere to a "Bach-adjacent", Baroque aesthetic.

4.1.3 Design choices and creative constraints

To fully embrace the design of NONOTO as aembrace the design of NONOTO as a direct, reactive interface with few technicalities, we decided to not allow individual regeneration of voices, only enabling to regenerate them all together in group fashion. Individual regenerations would actually be easily implemented with e. g. the DEEPBACH backend, by restricting the (blocked) Gibbs sampling strategy used

6 https://github.com/djipco/webmidi
7 https://github.com/Ableton/link/

by the model to only resample the content of the selected voices, but we decided against this to ensure a fluid usage. Furthermore, we wanted to leave enough flexibility *to the model*, embracing the idea of co-creative AI: we believe that if a user decides to use NONOTO, they should do so with the desire of letting the underlying models "express" themselves to some extent, which would be compromised by enabling micro-editions at too small scales. This would stand in contradiction with the general design decisions of higher-level interactive representations underlying this work. These decisions are actually also in line with the principles underlying Gibbs sampling, helping avoid moving too far away from the appropriate sampling distribution of the model, in that convergence towards the training distribution in Gibbs sampling is ensured only when individual local Gibbs resampling operations are performed homogeneously over the data.

As is the case in the other two interfaces presented in the thesis, instantaneous undo and redo is always available on screen, via the two arrows displayed to the left (undo) and to the right of the application's title. The undo button can also be used to cancel regeneration requests midflight, reinforcing the reactivity of the interface by following the "It's better to ask for forgiveness than for permission" principle, instead of relying on a Generate button to trigger generations.

Additionally, in order to further maximize screen real-estate utilization (especially key to making the interface effective on mobile devices), the application's title doubles as a button that can be clicked to trigger the "resetting" of the interface, requesting the backend generative model to generate a brand new sheet from scratch, which I believe could be beneficial to easily escaping the writer's block. Since this generation is stochastic, the button can be clicked several times in succession to obtain always new starting points, in what I personally see as inserting a new coin into the slot machine. Finally, the bottom controls (playback control, tempo, as well as advanced controls such as MIDI-Outsettings and Ableton Link setup...), more rarely used, can be toggled away entirely to only display the sheet and the two, more frequently used, undo and redo buttons.

4.2 RELATED WORK

In this section I propose a brief overview of related tools for interactive, symbolic music generation. I focus first on approaches that pre-dated

my work on the NONOTO interface in 2018 and 2019, highlighting the limitations in terms of reactivity and control of then-existing tools. I then present the COCOCO interface by Louie et al. [73], a conceptual successor of the influential Bach Doodle by Huang et al. [57] and a close cousin to NONOTO, released a few months after our own work. COCOCO focuses on adding controllability, or as they call it *steerability* to AI-assisted music generation, and represents an alternative approach to NONOTO, focusing less on reactivity and more on control. I discuss the specificities of both approaches.

4.2.1 Prior work on interfaces for symbolic music generation

The proposed system is akin to the FlowComposer system [86] which offered to generate sheets of music by performing local updates (using Markov Models in their case). However, this interface does not exhibit the same level of interactivity as ours since no real-time audio nor MIDI playback is available, which limits the tool to offline settings in a generate and curate fashion, making for a less spontaneous user experience.

The tools proposed by the Google Magenta team as part of their Magenta Studio effort [92] are more aligned with our aims in this project: they offer a selection of Ableton Live plugins (developed with Max for Live) that make use of various symbolic music generation models – for rhythm as well as for melody [59, 93]. Similarly, the StyleMachine, developed by Metacreative Technologies[8], is a proprietary tool that allows to generate midi-tracks into Ableton Live in various dance music styles using a statistical model trained on different stylistic corpora of electronic music. Yet these tools differ from ours in the generation paradigm used: they offer either continuation-based (the model generates the end of a sequence given the beginning) or complete generation (the model generates a full sequence, possibly given a template), thus breaking the flow of music on new generations. We believe that this limits their level of interactivity as opposed to local, inpainting-based models as ours, as mentioned previously. In particular, it hinders their usage in live, performance contexts.

Closest to our work on NONOTO is the COCOCO interface by Louie et al. [73], which was published after the original release of NONOTO and offers on a very similar approach, yet with strong differences in terms of UX design choices. I report in the following section some

8 https://metacreativetech.com/products/stylemachine-lite/

findings of a valuable UX study led by the authors of that work and discuss the differences in both interfaces.

4.2.2 The COCOCO interface

The Cococo interface (depicted on Figure 4.4), proposed by Louie et al. [73] in the continuity of the Bach Doodle presented in Section 3.3.2, aims at addressing the limitations in steerability exhibited by the Bach Doodle interface. It does so by enabling note-level editing as well as iterated inpainting operations at the note-level, which makes it highly relevant in the context of this PhD. Yet it does not aim for reactivity, instead offering to precisely parameterize the desired output via sliders prior to manually triggering generations. More precisely, features introduced by Cococo include the ability to obtain multiple (typically 3) propositions for each generation operation and let the user select the one they want to keep (zone F in Figure 4.4). Another feature is the introduction of sliders (zones D and E) which enable to steer the model towards specific styles and moods in order to obtain, e.g., respectively sad or joyous music. The sliders also enable controlling sampling parameters of the generative model to obtain more unexpected or, conversely, more conventional outputs.

In this perspective, by allowing to edit individual notes by hand and to perform inpainting operations on arbitrary zones (as opposed to only on slices of all voices simultaneously as in ours), Cococo offers more fine-grained control that our own approach with NONOTO, yet it does so at the cost of a potentially less reactive user experience. Indeed, because of the reliance on auxiliary sliders and the *Generate* button, any given individual editing operation requires more physical user actions. These additional control also make for a more densely packed interface, as shown on Figure 4.4: each of the letters from A to H describes conceptual interactive zones of the UI (these zones are presented in detail in the Cococo paper [73]). This arguably makes for a higher cognitive load than NONOTO, which ultimately consists of just the clickable overlay boxes and the selector to vary the size (granularity) of these boxes. It might also make it hard to conveniently transfer such an interface to mobile devices. Furthermore, Cococo does not provide any means of directly integrating within DAWs, potentially hindering its adoption by musicians.

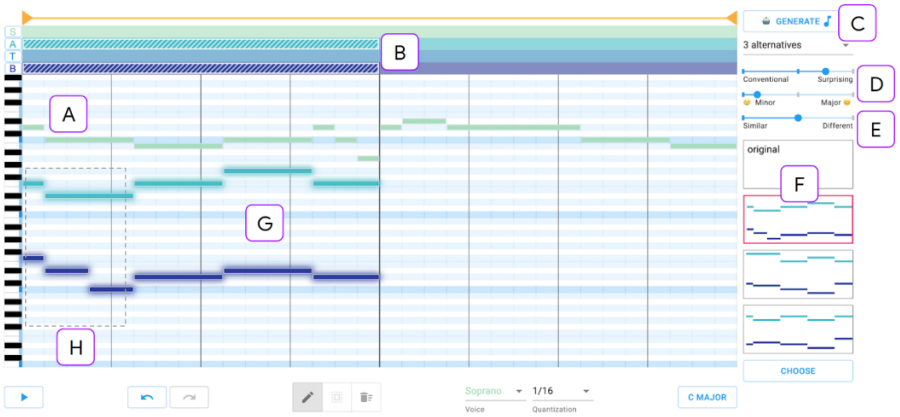


Figure 4.4: The Cococo interface. (Image taken from [73].)

THE COCOCO UX STUDY The authors of this paper conducted a user study with N = 21 subjects to evaluate the reception of the Cococo interface in terms of the gain in appropriation and adoption brought by the specific additions introduced in Cococo to improve control. Users were asked to create some music either one of two different versions of the Cococo interface: the standard one, including all steering features, and a so-called "conventional" version, which striped away these various features to get closer to what the authors depict as the standard feature set of existing interfaces for AI-assisted music creation. Users when then asked to fill a short survey on their experience during this task. According to the results of this study, the users largely preferred working with the steering-enabled, full-featured interface. These results somewhat contradict our design choice of building a minimalistic, "buttons-free" interface by, e.g., not enabling per-voice edition. This paper thus represents an interesting point of reference and could spawn further developments to NONOTO as well as motivate a specific user-study to re-assess the findings of the user study in the context of our reactive NONOTO interface. Of particular interest would be a situated study when NONOTO is used in synchronization with Ableton Live though Ableton Link for jamming or live music performance, a context in which the fluidity of the interactions is quintessential and could potentially be disrupted by the added steering controls of Cococo. Taking cues from the results of the study and designing and implementing an efficient system for per-voice inpainting in NONOTO could provide an opportunity for an interesting UX study to confirm or contradict those of Louie et al. [73].

4.3 TECHNICAL DETAILS

Our framework relies on two elements: an interactive web interface and a music inpainting server. This decoupling is strict so that researchers can easily plug-in new inpainting models with little overhead: it suffices to implement how the music inpainting model should function given a particular user input. We make heavy use of modern web browser technologies such as the Web MIDI and Web Audio API, making for a modular and hackable code-base for artists and researchers, allowing e.g. to edit the interface to allow for some particular means of interaction or to add support for some new metadata specific to a given corpus.

4.3.1 Interface

The sheets are visually rendered in SVG format in the browser via OpenSheetMusicDisplay [124], an open-source TypeScript library for rendering MusicXML sheets with vector graphics. Using `Tone.js` [75], an open-source JavaScript library for real-time audio synthesis and musical scheduling, I augmented OSMD with real-time audio playback capacities, allowing users to preview the generated music in directly from within the interface. Furthermore, the audio playback is uninterrupted by re-generations, enabling a truly interactive experience, ready for jam-like usage. This enabled a shift away from traditional generate-and-curate approaches in which some content is first generated, *then* the user listens to it and decides whether or not they like it, *then* can regenerate either all of it or just a part, with the playback getting interupted and restarted on every single, albeit local update to the generated music.

To make this on-the-fly regeneration UX as smooth as possible, I adapted my `Tone.js`-based ad-hoc MIDI-sequencer to perform all user-triggered updates to the music scheduled for playback in an *in-place* fashion with every inpainting operation: the audio engine only removes previous and adding new notes of the inpainted zone. Prior to implementing this optimization, the playback would stutter on each inpainting operation – which would naively trigger an unnecessarily heavy full cancelation and re-scheduling of the whole sequence on each localized update, making for an arguably sub-par experience, especially on less-powerful hardware such as mobile phones. Thanks to

these in-place updates, the tool runs smoothly on desktop computers and common mobile devices.

4.3.2 Generation back-end and communication

For better interoperability, we rely on the open MusicXML standard to communicate scores between the interface and the server. The HTTP-based communication API then just consists in two commands that we expect the backend server to offer:

- An `/inpaint`[9] command which takes as parameter the position of the interval to re-generate. The client triggers a call to this API endpoint whenever the user clicks on one of the interactive overlay boxes displayed over the sheet, and sends to the server the associated time-range and the current sheet as well as potential metadata guiding the generation model such as the positions of fermatas in the NONOTO-DeepBach setup or the current user-selected chord annotation constraints when used with models supporting those. The server is then expected to return an updated sheet with the chosen portion regenerated by the model using the current musical context.
- A `/generate` command which expects the generation model to return a fresh new sheet of music in the MusicXML format. This context-free command can be used to initialize a session, or to obtain new starting points when one reaches a creative block.

Note that these two API methods are purely stateless. This has significant advantages for deployment, enabling easy, fault-tolerant deployment via stateless container images. I describe this in Appendix A, where I provide more details on the front-end and back-end infrastructure used for the prototypes presented in this thesis.

MUSICXML TO MIDI CONVERSION In order to render the generated sheet music within NONOTO, we must rely on the MIDI format, since MusicXML does not appear to be easily supported as an input format by the usual JavaScript-based music sequencers we found, and in particular can not be parsed by `Tone.js`. Since I did not manage to find a JavaScript MusicXML-to-MIDI converter, I ended up performing this conversion on the server (with the `music21` Python package), which

9 Currently called `/timerange-change` in the implementation due to legacy reasons.

means we expose an additional `musicxml-to-midi/` endpoint on the inference server.

4.4 SYNCHRONIZATION WITH ABLETON LINK

Ableton Link provides a mechanism for distributed synchronization of musical sources and is distributed as a C++ library, under a dual GPLv2+ or proprietary license. It is described by its authors as:

> "A technology that synchronizes musical beat, tempo, and phase across multiple applications running on one or more devices. Applications on devices connected to a local network discover each other automatically and form a musical session in which each participant can perform independently: anyone can start or stop while still staying in time. Anyone can change the tempo, the others will follow. Anyone can join or leave without disrupting the session."[10]

In the following section, I briefly discuss how I integrated those with the `Tone.js`-based interactive audio engine developed for NONOTO. I then give some details on how I could integrate the `node-abletonlink` package with the NONOTO frontend, since `node-abletonlink` cannot be run directly within the browser but rather needs to have direct access to hardware APIs for running its communication processes. I therefore ended up deciding to package NONOTO as an Electron application, bundling together the frontend and the `Node.js` runtime, offering an easy-to-use, one-click executable solution for end-users.

In short, Ableton Link exposes two core features:

1. Synchronization of position in the playback grid (called "phase" in the Ableton Link API), enabling to *start in sync with other sources*,
2. Synchronization of tempo changes, ensuring that currently playing sources can *stay in sync at all times*.

That is, when pressing play in NONOTO, the interface waits for Ableton Link to signal that the other tracks are reaching the beginning of a new measure and only then adequately starts playing with them, right on the beat. Whenever, then, a tempo change occurs, e. g. within Ableton Live, the playback tempo in NONOTO is automatically updated. Finally, I devised and added a periodic phase-locking mechanism that

10 Introduction of the official Ableton Link README file at https://github.com/Ableton/link/#ableton-link

automatically re-aligns the playback position of NONOTO with the Ableton Link phase, in order to avoid drifting away from perfect synchronization over time. Such phase drift would otherwise frequently happen due to unavoidable instabilities in the non-real-time oriented web-based audio engine, such as slight frame-drops, often due to the web engine randomly dropping priority of the audio engine's process. Thankfully, this hack-y phase-locking solved this issue and made this drift unnoticeable, ensuring pleasant playback stability. The realignment step simply jumps to the current Ableton Link phase once every measure, which remains unnoticeable since the sequencer is only outputting MIDI commands, thus not involving audio frame-skips.

I know discuss the more technical aspects of this integration.

4.4.1 Integrating the `node-abletonlink` bindings

The integration of Ableton Link within NONOTO is made possible first and foremost by `Node.js` bindings, distributed as an open-source package on GitHub by user Ishii [125] under the name `node-abletonlink`. `Node.js` is the server-side engine for running JavaScript code. It being server-side code means that these bindings require access to some hardware specific APIs that the web browser does *not* expose and the Ableton Link server must therefore be run in a dedicated server-side process. As such, it could not be readily integrated into the NONOTO interface and required further consideration.

The initial solution I designed for connecting the NONOTO frontend to the Ableton Link server was therefore to separately run the `node-abletonlink` bindings and the frontend, and connect the two via an ad-hoc `socket.io`-based API. This had the stark disadvantage of requiring potential users to start a local `Node.js` server manually before initiating a session, which we cannot expect non-tech savvy users to be able to do.

The second and much more convenient solution came from packaging NONOTO as an *Electron* application [122]. Electron is a cross-platform, open-source `Node.js`- and JavaScript-based platform that "[...] embeds Chromium and `Node.js` to enable web developers to create desktop applications" as per its official description. This solution allowed me to conveniently package together, into a single executable application, both the frontend side of NONOTO (the sheet visualization, the on-the-fly MIDI sequencer) and a copy of the `node-abletonlinkNode.js` bindings. Electron then provides the custom Inter-Process Communi-

cation (IPC) module, a channel for setting-up efficient asynchronous and bidirectional communication between the `Node.js` engine and the user-facing interactive frontend code running in the embedded Chromium engine. This solution has the advantage of being fully transparent to the user, who can them simply retrieve the built executable for their platform (Linux, macOS, Windows), turn on Ableton Link synchronization through a click of a button *in the frontend* and start jamming right away.

PHASE LOCKING AND LATENCY COMPENSATION Proper support for Ableton Link, however, did pose some technical issues, since tight synchronization of multiple, independent sources requires performing *latency compensation*, in order to properly align them in time. Indeed, each audio processing pipeline made up of any given software and hardware combination inevitably introduces slight computation delays, which are *platform* and *software dependent* and must be accounted for and properly compensated if perfect synchronization is to be obtained [44].

Official Ableton Link implementations are expected *to perform such compensation*

Thankfully, I could partly circumvent this issue in that the NONOTO use case does not require that we perform latency compensation at the *audio level*. This would have involved computing the latency of the hardware, audio generation pipeline. As NONOTO however only outputs symbolic, MIDI signals, this could be avoided. Nevertheless, the `Tone.js`-based sequencer engine needs a small look-ahead for it to perform optimally, that is, a short window of anticipation during which it can process and prepare the incoming events. This look-ahead [11] introduces, by definition, some latency in the outputs produced by NONOTO. By experience, the stability of the `Tone.js` scheduling engine quickly degrades when using too small a look-ahead, so setting the look-ahead to zero was not an option. Yet, this delay made it impossible to properly synchronize NONOTO with Ableton Live. I eventually managed, after several failed attempts at hacks over the course of the PhD, to correct this inherent temporal delay. To do so, I simply had the sequencer engine automatically shift the whole scheduled sequence forward in time by exactly the duration $t_{lookahead}$ of the look-ahead. This time-shift naturally compensates the delay by scheduling all events *in advance*, effectively shifting them by $-t_{lookahead}$. Since the events are then emitted with a delay of $t_{lookahead}$, the resulting perceived latency is null.

11 Typically, in the order of 0.1 s as advocated by the official `Tone.js` documentation, https://github.com/Tonejs/Tone.js/wiki/Performance#contextlookahead.

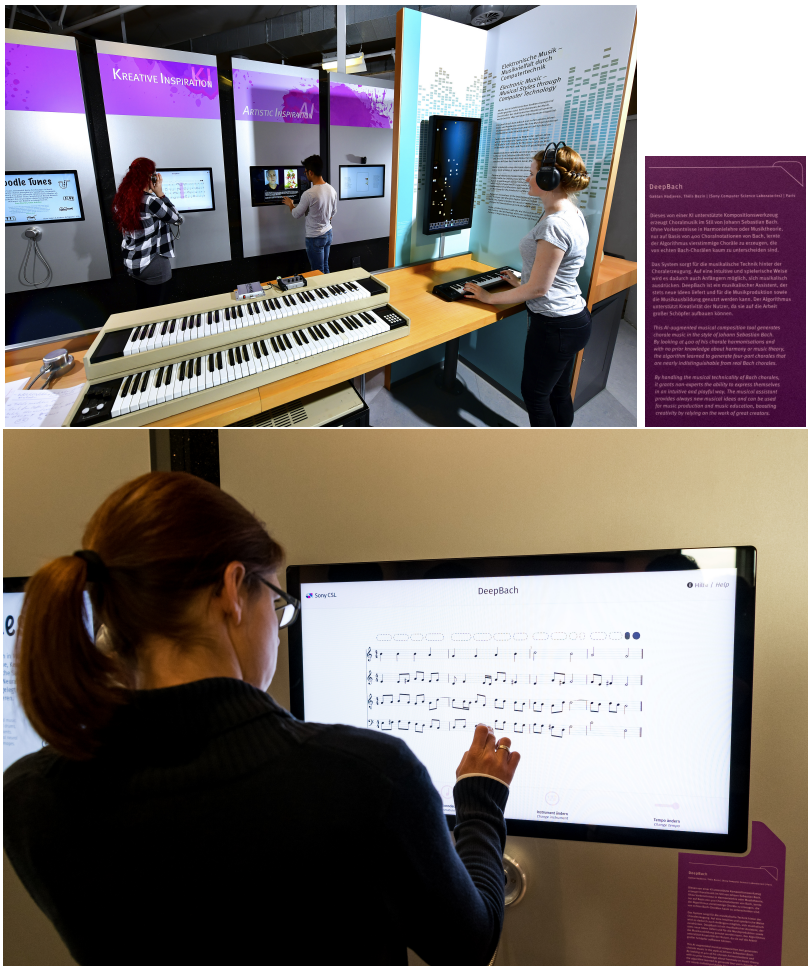


Figure 4.5: Pictures of the NONOTO interface as exposed at the Heinz-Nixdorf Museum of Computer Science, Paderborn, Germany, October-December 2018. Pictures courtesy Dr. Doreen Hartmann.

Note, however, that this hack nevertheless has the negative side effect of skipping, on every initial playback of the generated content, the very first events of the sheet, which are effectively shifted away *before* the beginning of the sequence. These events are nevertheless properly played on every successive, looped playback of the sequence, by scheduling them at the *very end of the sequence*, effectively triggering them on the start of it, after compensation of the look-ahead. This aspect would remain to be improved[12].

Figure 4.6: NONOTO demo booth at NeurIPS 2018, with co-author Dr. Ashis Pati. Photo by Dr. Gaëtan Hadjeres.

4.5 DISSEMINATION

NONOTO, along with the DEEPBACH model for which it was initially designed, was exhibited in Q4 of 2018 at the Heinz-Nixdorf Museum of Computer Science in Paderborn, Germany, as part of an exhibition curated by Dr. Doreen Hartmann on recent progresses in the field of AI. During this exhibition, visitors could interact with the interface through a large touch-enabled display. I show some pictures of this exhibition on Figure 4.5.

NONOTO was also presented as a live demo at NeurIPS 2018, in collaboration with co-author Ashis Pati, who worked with Gaëtan Hadjeres on building a Folk-music model that was part of the context-aware editing models used to prototype NONOTO. A photo of the booth is shown on Figure 4.6. For the demo, NONOTO was showcased in conjunction with Ableton, controlled via the Ableton Push hardware controller, and using a touch-enabled projection device. The projector displayed the NONOTO on the table and users could touch regeneration zones directly on the table to trigger inpainting operations. We would then simultaneously control synthesizer parameters as well as drum loops sequencing within Ableton, using the Ableton Push hardware device visible on the table. Attendees of NeurIPS – with various academic or industrial backgrounds, with or without experience in music performance or production – generally reacted positively to this live demo, highlighting the easy interaction offered by the interface and the large possibilities of customization offered by the integration with Ableton. Attendees valued the ease-of-use of the proposed interface,

12 And it could very well be that my lack of mastery of the techniques for latency compensation made me miss an obvious solution to this issue...

mentioning that it made the idea of co-composition and AI-assisted music more approachable and less fearsome to them. Some attendees also expressed interest in integrating such interactive and accessible techniques in music pedagogy.

Finally, NONOTO was also selected and showcased in an industrial setting as part of the 2018 Sony CSL Open House at the Sony CSL Tokyo laboratory. Attendees included engineers, researchers and top-executives from Sony, as well as guests from the academic, industrial and cultural sectors in Japan.

4.6 CONCLUSION: HOW TO GO FORWARD FROM HERE?

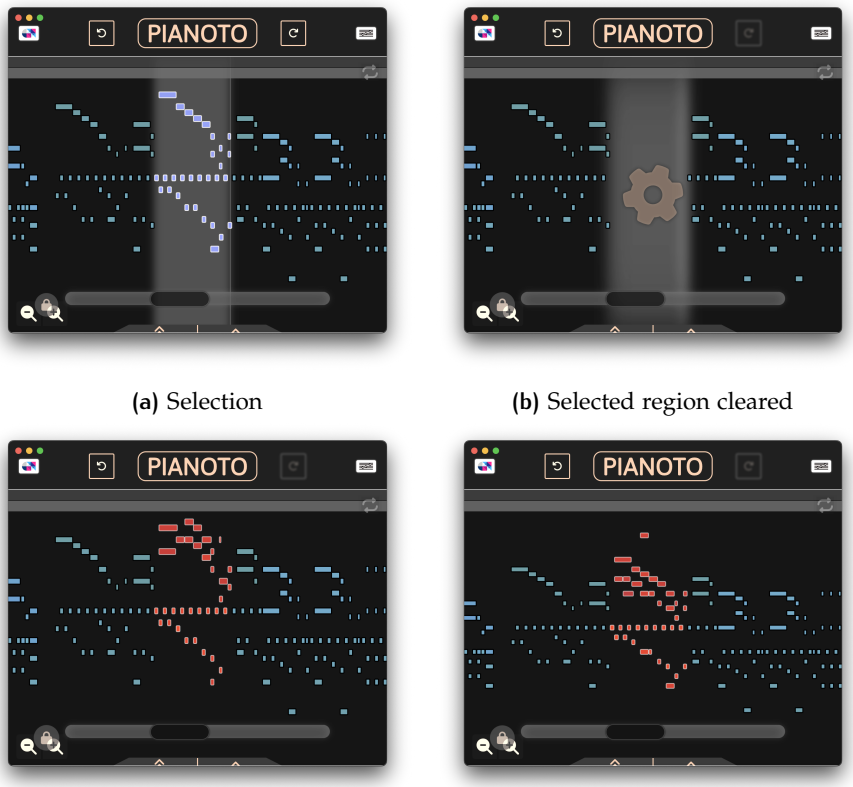
I have introduced NONOTO, an interactive, open-source and hackable interface for music generation using inpainting models. We invite researchers and artists alike to make it their own by developing new models or means of interacting with those. This high level of hackability is to a large extent permitted by the wide range of technologies now offered in a very convenient fashion by modern web browsers, from which I draw heavily. Ultimately, I hope that providing tools such as ours with a focus on usability, affordance and hackability will help shift the general perspective on machine learning for music creation, transitioning from the current and somewhat negative view of "robot music", replacing musicians, towards a more realistic and humbler view of it as A.I.-*assisted* music.

The presented interface however exhibit crucial limitations in its usability by musicians, as well as in the richness of its outputs, due to the constraining reliance on sheet music. This limitation made sense back this tool was first developed, since the state of the art for music generation was still very much restricted to quantized representations and sheet-music. The following chapter proposes, with the PIANOTO interface, to move away from these restrictions by making use of recent advances in symbolic generative modeling.

5 PIANOTO

The Piano-Roll as an Instrument

> – Ils n'ont pas demandé à vivre, et voilà qu'on leur apprend le piano en plus, que voulez-vous.
>
> in *Moderato cantabile* (1958)
> Marguerite Duras



(a) Selection **(b)** Selected region cleared

(c) Inpainting result #1 **(d)** Inpainting result #2

Figure 5.1: Using PIANOTO to generate several variations (5.1c and 5.1d) of a sequence through repeated inpainting of the same zone. The music is from the 3rd Movement of Mozart's Symphony No. 41 in C major. The model successfully picks up on the left hand motif and offers melodic variations on the right hand.

RATIONALE

In this chapter, I build upon the ideas underlying the NONOTO presented in the previous chapter, but propose to increase the expressivity and flexibility of this co-creative interface by moving from relying of a quantized representation of music in the sheet format to the expressive, unquantized MIDI format. This shift directly follows recent advancements in neural modeling, namely the learning of compact, discrete intermediate representation with Vector-Quantized Contrastive Predictive Coding (VQ-CPC) and the use of Transformer-based architectures to subsequently perform interactive regeneration over these compact representations by means of inpainting. In terms of interface and interaction design, this work is an attempt to directly address crucial issues of the piano-roll described in Chapter 2.

I gently invite the reader to watch demo videos of this tool before delving into the description contained in this chapter. First, an introductory video[1] to the general usage of this tool, where I create a short piano excerpt absolutely from scratch in an iterative fashion. Second[2] is an example, recorded for the 2022 ISMIR Late-Breaking Demos session, of using PIANOTO to transform an expressive, audio-recorded piano performance, by means of the integrated piano transcription capabilities (powered by Google Magenta's ONSETS AND FRAMES). The latter demo video also includes an example of using PIANOTO on a mobile phone to create, from scratch, some very "Debussean" music.

5.1 INTRODUCTION

The piano-roll has been the *de facto* standard representation for melodic and harmonic content in DAWs for decades, yet, as discussed previously, direct manipulation of those requires expert knowledge of music theory to begin with, and additionally becomes physically impractical when switching to the smaller, touch-based screens of modern mobile devices, too coarse for the precision required by micro-timings and the unforgiving discrete placement of pitches. As such, in spite of the possibility to e. g. position notes flexibly in time over those grids in an unquantized fashion, piano-rolls tend to be used in a strictly quantized fashion, with notes automatically slices and aligned to a fixed

1 https://youtu.be/n17PxtmZZQM
2 In the *Video* section of the prototype's page on the ISMIR'22 website: https://ismir2022program.ismir.net/lbd_395.html

subdivision of the metric musical time. This quantization helps alleviate the aforementioned impracticalities of the otherwise too precise interactions, but it dramatically limits the expressivity of the resulting music. Such a quantization is at the very least the default setting in a majority of DAWs, as discussed in an essay by Faber [40]. Imbuing these interfaces with machine learning and offloading their precise and error-prone aspects to style-adaptive AI assistants may allow the design of more intuitive interactions whilst maintaining a high level of control, helping lower the cost of entry to composition for novices and offer stimulating new creative tools for professional musicians.

PROPOSED APPROACH In this chapter, I introduce PIANOTO, a touch-ready, responsive web interface for creating expressive piano performances through AI-assisted inpainting, all via simple swipe operations. This open-source, model-agnostic prototype is designed with both novice and expert users in mind, for usage either as a standalone tool or in conjunction with existing DAWs, on desktop or mobile. The compatibility with mobile devices is made possible by *directly augmenting the piano-roll itself* to make it conveniently interactive, instead of adding additional, external controls in the form of knobs or buttons. Screen space utilization is thus maximized and the piano-roll visualization becomes a welcoming, easy-to-use instrument. This is done by inserting a generative model between the low-level, precise representation of the piano-roll and the user. Users can select *time slices* on the piano-roll that are then regenerated by an *inpainting*-enabled generative model, that is, a stochastic model of the form $p_\theta(\texttt{note}_t|\texttt{notes}_{<t}, \texttt{notes}_{>t})$, able to generate new, *meaningful* notes, accounting for the past and future notes. The model takes care of inserting micro-timings and expressive velocities, without requiring any overly precise interactions from the user. This interactive process is shown on Figure 5.1.

5.2 BACKGROUND: PIA

The PIANOTO interface builds upon the PIA model by Hadjeres and Crestel [49], itself part of a larger field of expressive music modeling, with examples including the Music Transformer architecture of Huang et al. [59]. The PIA model is an example of the concepts discussed in this thesis, combining the design of a compact, intermediate representation and the training of powerful, context-aware auxiliary

autoregressive generative models (realized as efficient Transformers) over those compact representations, ultimately enabling direct interactive generation of expressive piano performances.

HIGH-LEVEL REPRESENTATION: STRUCTURED MIDI ENCODING The intermediate, higher-scale representation proposed with the PIA model aggregates MIDI note events (otherwise depicted as two separate `NoteOn` and `NoteOff` events) into single 4-uples containing: *Pitch*, *Velocity*, *Duration* and *Time Shift*. The Time Shift was a crucial innovation of this models and represents the delay between the note represented by a given such tuple and the following note. Accordingly, this *Time Shift* can be zero when multiple notes are played at the exact same time. This yields an aligned representation of performances as sequences with a strict chronological order and removes the problematic interleaving of `NoteOn` and `NoteOff` events, a spurious interleaving which otherwise dramatically hinders statistical modeling. The intermediate representation is therefore simple 1-dimensional, temporal sequences of abstract events, which can be manipulated as such. This therefore removes the need for the user to explicitly set the precise timing, pitch and velocity details, since those are all abstracted away and handled by the model.

EFFICIENT TRANSFORMERS FOR INTERACTIVE GENERATION The authors of PIA subsequently use this condensed representation to train efficient, context-aware Transformers, furthermore introducing various optimizations in the representation of positional embeddings and autoregressive sampling scheme that enable to scale the models for high modeling capacity yet maintain efficient inference times.

INTERACTION AND THE MAX4LIVE PLUGIN Ultimately, the conceptual interaction enabled by this architecture consists in selecting a temporal slice and prompting the underlying model to insert some events into it, taking into account the surrounding note events. All fine-grained details are handled and resolved by the model, making for a simple and fluid interaction.

Along with the original PIA paper, the authors proposed an interface for integrating its use within Ableton Live in the form of a Max4Live patch, visible on Figure 5.2. This plugin allows to select zones on the Ableton Live piano-roll, and click on the unique displayed "Generate" button to trigger inpaintings. Unfortunately, using this device requires

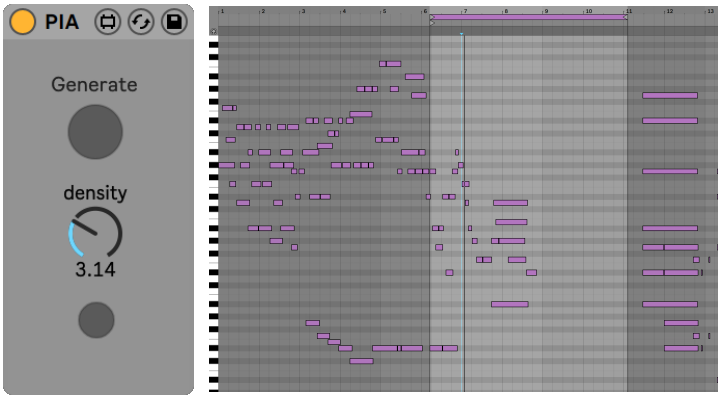


Figure 5.2: The PIA Max4Live device (left) and the Ableton Live piano-roll (right).

constant toggling between the piano-roll view and the devices view of Ableton to trigger inpainting operations (which cannot be displayed together at the same time on screen), making for an arguably sub-optimal user-experience. This can be attributed to the impossibility of directly extending the piano-roll component in Ableton Live, forcing the reliance on hacks as done with this plugin.

Through the present work, I sought to improve on the design of this plugin and fully realize the potential of inpainting-based models of expressive music for the construction of novel interactive modalities with piano-rolls.

5.3 PROPOSED INTERFACE

We propose PIANOTO, a *model-agnostic web interface* that replaces the precise note-level interactions of traditional piano-rolls with simple temporal selection, as shown on Figure 5.1. Screenshot of the interface in desktop and mobile display are visible on Figure 5.3.

When operating on imported music (either recorded via MIDI-In, loaded as a MIDI file or transcribed from an audio file, as described below) the interface displays notes coming from this initial material in blue as opposed to displaying notes created by PIANOTO in red. This is driven by two ideas. First, it underlines the function of PIANOTO as a co-creative assistant, by visually signaling what part of the current performance was co-created with the model and what part originates from the source material. Second, it directly seeks to provide visual assistance for editing material. Since the original material guides the

Figure 5.3: The PIANOTO interface, used both on desktop and on a mobile device. By not relying on external controls such as "Generate" buttons, the whole display focuses on the actual, musical content.

generative model, visualizing at any moment what portion of the current, co-created performance still directly comes from the original material is an indication of whether this source material is still strongly guiding the generation. Indeed, the more the piece gets edited, the less it is clear whether the original material is still providng contextual, stylistic information.

5.3.1 From novices to professionals

Inpainting enables a wide range of use cases, from millisecond-scale edits to macro-scale (re)generations. Unconditional generation, crucial for novices to obtain valuable initial content, can be achieved by repeatedly generating material from scratch until obtaining a satisfactory starting point. Conversely, expert users can import their own performances as MIDI files or directly record themselves via MIDI-In and transform those. Users can also drag-and-drop audio files to create personal, expressive variations of existing performances, via built-in transcription through Google Magenta's ONSETS AND FRAMES model [55, 126]. Since every generation step then accounts for the evolving piano-roll content, the users can progressively shape it to their taste by keeping zones which they find valuable and regenerating others. The context ultimately reflects the user's style, accounted for by the model. Inpainting thus enables a form of *on-the-fly user-adaptation* of large models, without requiring expensive retraining.

5.3.2 The piano-roll as an instrument

Our aim with PIANOTO is to turn the piano-roll from a static, expert-only *interface* into a playful yet rich *instrument*, as embodied by the "Low entry fee with no ceiling on virtuosity" philosophy of music interface design pioneers Wessel and Wright [114]. The interface is reactive, in that selecting a temporal zone by click-and-drag or swipe on the piano-roll *immediately* triggers an inpainting. Furthermore, by filling any selected zone with *multiple* events, we turn the piano-roll's core interaction modality from rigid, purely deterministic note-by-note edition into rich, spontaneous and always renewed *one-to-many interactions*, a factor enabling the design of rich musical interfaces, as discussed in the seminal 2002 NIME paper "The Importance of Parameter Mapping in Electronic Instrument Design" by Hunt, Wanderley, and Paradis [60].

THE REGRETS-FREE COMPOSER This reactive approach is also driven, as in NONOTO, by the "It's better to ask for forgiveness than for permission" principle, and instantaneous *Undo* and *Redo* are always available on screen, either on desktop or mobile, with the undo button (again) doubling as a cancel operation for ongoing (potentially long) re-generation operations. Here, the application's title doubles as a button that erases and resets the current content of the piano-roll, allowing to restart a new creation from scratch.

As well as through the classical keyboard shortcuts on desktop.

BACK TO THE VIEWPORT! Crucially, all interactions happen directly *on the musical representation*, rather than through external controls (e.g., there is no "generate" button). This is in line with recent work, e.g. in the field of 3D modelling: Michel and Boubekeur [76] argue for moving interactions "back into the viewport", for instance by visually dragging parts of a 3D model to move or resize them instead of relying on external sliders.

5.3.3 Batched processing in PIA and visual animations

Care was given to bring the generation process to life through the introduction of animations within the interface. These animations build upon an initial idea and implementation by Hadjeres and Crestel

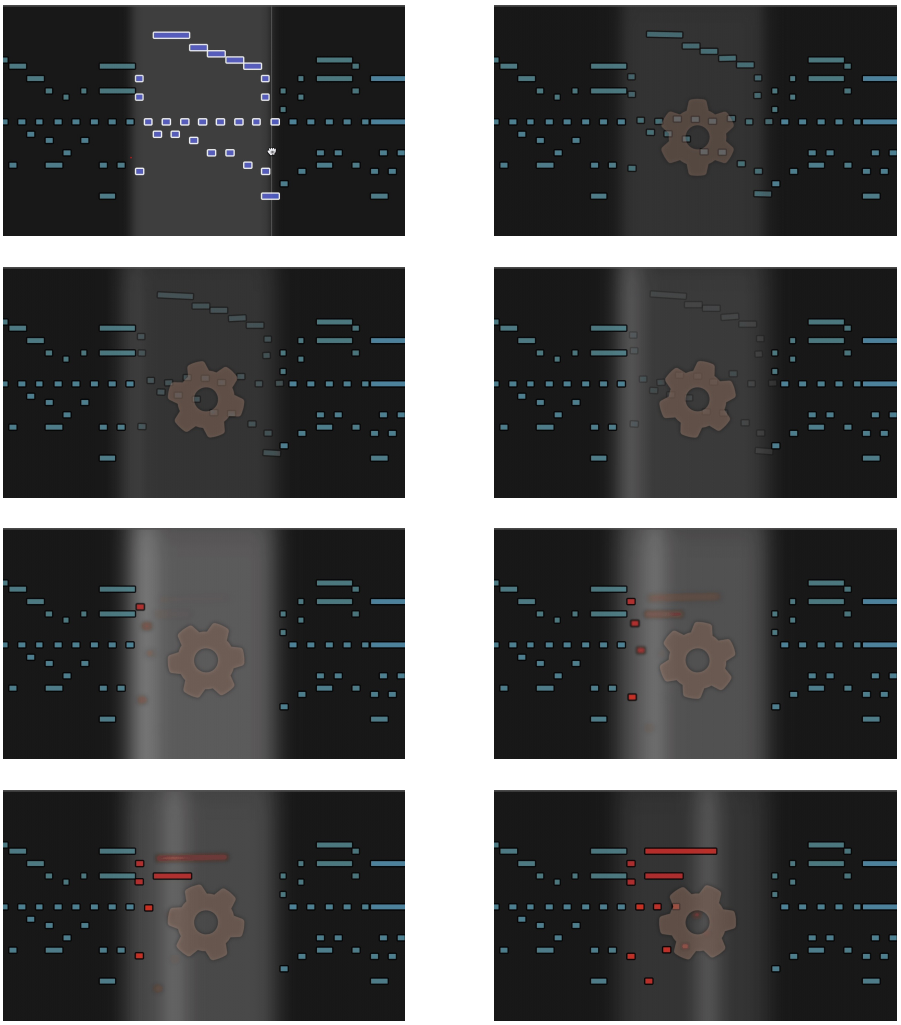


Figure 5.4: Frame-wise decomposition of the visual animations included in PIANOTO for note removal and note insertions during inpainting operations. Notes are translated and rotated upwards in addition to being blurred away when getting removed and, conversely, progressively appear from below and come into place with a magnetic feel upon insertion by the AI model.

[49], aided by Adrien Laversanne-Finot[3], as part of the initial Max for Live PIA plugin.

NOTE-BY-NOTE REVEAL IN THE MAX4LIVE PLUGIN When using the PIA model, newly generated notes from the backend API arrive to the client in successive batches of approximately 10 to 15 notes. This batched processing was decided upon to improve the stability of the API by avoiding to flood it with micro-requests for generating a single note. For the Max for Live plugin, this meant that during generation of a zone, the plugin would now and then suddenly print a new, medium-sized batch of notes to the Ableton Piano-Roll before becoming silent for a while more, waiting for the next batch of notes. In order to avoid such an intermittent visualization the authors of that plugin decided that received notes would be rendered in succession by inserting an artificial delay between each note insertion to the piano-roll. Effectively, this gives the impression of notes arriving to the client sequentially, thus making for a smoother, more progressive visual experience and highlighting the autoregressive process at play in PIA.

... AND ITS EXTENSION IN PIANOTO The animations in PIANOTO, displayed on Figure 5.4, similarly focus on the addition and removal of notes as part of the inpainting operations and, in addition, seek to imbue the interface with a feeling of livelihood. Thus, in addition to the sequential insertions with artificial visual delay, I added progressive blurring and de-blurring effects as well as smooth opacity transitions, so that, rather than abruptly erasing and printing notes onto the viewport, the notes are brought softly in and out of view. Additionally, translation transitions were introduced to move erased notes up the viewport simultaneously to blurring them out, with an underlying metaphor of sending these notes out to a conceptual *firmament* of notes bygone. Similarly, newly generated, incoming notes arise from below their final, real position on the piano-roll and appear from an initial blurry state, as though emerging from a void of silence, brought by PIA and the hidden power of AI into their rightful position.

Waxing poetic a bit here.

RECEPTION AND DISCUSSION These animations were very positively welcomed by collaborating musicians CHATON and Whim Therapy as bringing some identity and a sentiment of magic to the

3 Developer for the Max for Live PIA plugin.

interface, with Whim Therapy writing[4] that "The midi notes drawing in front of me feels like magic".

A limitation to this design, however, is that animating blurs is rather taxing on browsers' graphic rendering engines and, in particular, seems to introduce some lag when using PIANOTO on mobile phones. This can however be circumvented by disabling the use of blur effects and only relying on opacity animations on those devices. This slightly reduces the overall visual effect, but ensures full efficiency and avoids encountering any frame-rate drops. Proper optimization of this rendering is therefore still ongoing.

5.3.4 Backend: Model-agnostic, but batteries included

The PIANOTO interface is *model-agnostic* and we encourage researchers and practitioners to connect it to their own models, making it easy to turn any static model into an interactive instrument. PIANOTO was nevertheless designed around a new version[5] of the PIA model by Hadjeres and Crestel [49]. This Transformer-based model was trained for inpainting on GIANTMIDI-PIANO [67], a large corpus of solo classical piano performances obtained by automatic transcription of piano videos. PIA is able to predict pitches, velocities as well as note onsets and durations at a millisecond scale, enabling expressive timing. Interestingly, this new version of PIA also performs well on out-of-domain data and can adapt to more modern playing styles such as jazz, minimalism or pop music, ensuring that PIANOTO can be beneficial for a wide range of users. We note here a limitation: the free-form, solo-piano training of PIA can become impractical when users expect it to adhere to a fixed tempo, e.g. when composing a piano part to include into an existing piece. Imbuing this model with metronomic timing could therefore be the object of future work.

5.4 TECHNICAL BACKGROUND

Even without considering its inpainting-enhanced generative capabilities, PIANOTO offers a ready-to-use solution for single-instrument, polyphonic MIDI playback and recording in the browser, equipped with seek/loop, velocity-scaled display for improved readability and

4 As part of the user survey I conducted and which I report on in Chapter 8.
5 To be published. Model available in the interface's repository.

built-in MIDI-IN and OUT. PIANOTO uses the same audio engine as the NONOTO interface presented in Chapter 4 and the NOTONO interface presented in Chapter 7, based on the `Tone.js` Web Audio API framework [127, 75] and the `webmidi.js` library [121, 25].

PIANOTO furthermore integrates the ONSETS AND FRAMES transcription model and some UI elements from `Magenta.js` [126]. These UI elements, along with the associated `html-midi-player` [120] library, form the basis for the piano-roll visualization. PIANOTO expands on those and turns the static piano-roll of `Magenta.js` into an interactive instrument. We note here that the SVG-based implementation of the piano-roll as provided by `Magenta.js` proved beneficial for turning it into an interactive piano-roll, as each SVG node (and, in particular, each block representing a note) can be treated and styled dynamically as any other HTML element on the page, adding interactivity to this representation was very convenient.

5.5 RECEPTION AND DISCUSSION

PIANOTO is a very recent effort, and was actually, chronologically, the last prototype developed as part of this PhD. Accordinggly, it has not been used by many artists. Sony CSL collaborator Whim Therapy[6] nevertheless used it for some time, applying it in particular to the creation of variations of Vivaldi's *Four Seasons*. Answering a survey I conducted[7], Whim Therapy observed the following about PIANOTO.

To the agreement rating question "I could get valuable results with the tool quickly after using it for the first time", he gave an 8/10 rating, and to that "The more I used the tool, the more valuable the results I managed to get", he very positively agreed with a 9/10 rating. This validates the attempt of designing PIANOTO as an instrument, with a low floor, a low cost of entry, but with also a high ceiling, where learning to properly perform successive regeneration at various scales, thus controlling the context of the underlying model to guide its generations, can prove very more and more beneficial.

In open-ended questions, Whim Therapy is very enthusiastic about the design of the interface and about the unique creativity enabled by this prototype:

6 With whom we participated in the 2021 AI Song contest and whom I present in more details in Chapter 8

7 And which is, again, presented with more details in Chapter 8

Q. – *What (if any) were some things that you enjoyed about the tool?*

WHIM THERAPY – All the creative propositions it made.

Q. – *What (if any) were some things that felt surprising or exciting about the tool?*

WHIM THERAPY – The midi notes drawing in front of me feels like magic.

Q. – *How would you describe your relationship to the results obtained with the prototype?*

WHIM THERAPY – Very good!

Q. – *Do you feel like you could have achieved those results without the tool Why?*

WHIM THERAPY – I don't think so, as the ideas submitted by the tool were really different from the ones I have.

A BUG AND A MOTIF FOR FRUSTRATION However, Whim Therapy regretted that PIANOTO was not able to stick to an external tempo and grid (which is expected due to the current design of the tool, without any entrance for such input). He furthermore regretted however that the tool was not entirely able either to adhere to the internal tempo and grid of the MIDI files he imported. Part of it can be linked to the free-form nature of the underlying PIA model, due to its training on solo piano performances. Another issue nevertheless is also at play here, in the form of an – as of yet – unresolved issue in PIANOTO itself, which introduces weird timing issues in the generated content which lead to the inserted notes not properly reconnecting with the following, future notes when inpainting a slice. Practically, it sometimes sounds as if the inpainted sequence added by PIA were interrupted midway, due to reaching the end of the selected inpainting range, apparently not properly accounting for the end timestamp of said region. These timing issues are seemingly not to be found when using PIA along with its original Max4Live-based interface directly within Ableton Live, which tends to prove that the issue is not one of the core approach, jeopardizing the whole proposed approach, but a specific implementation problem of the PIANOTO prototype itself. A potential culprit could be in a mismatch between the underlying resolution of PIANOTO and the PIA model but attempts at addressing this have not yet proved successful. This issue is therefore still under active scrutiny.

Because of these issues, which forced him to manually re-edit the co-created variations with Ableton Live to re-align them, Whim Therapy

rated PIANOTOwith only a 3/10 agreement on the following statement: "It felt natural to integrate the tool within my usual music production practice". This is also reflected in his answers to the following open-ended questions:

> QUESTION – *What (if any) were some things that you felt were missing in the tool?*
>
> WHIM THERAPY – The ability to get in total sync with my midi files, so I don't have to edit it to put it back in rhythm.
>
> Q. – *What (if any) were some things that felt weird or frustrating about the tool?*
>
> WHIM THERAPY – The rhythm thing.

PERSPECTIVES Drawing from these remarks, it is clear that addressing these timing issues constitutes a crucial avenue for improvement of the PIANOTO prototype. Furthermore, as mentioned above and confirmed by Whim Therapy, integrating a mechanism for following dynamic tempo and timing within an extension of PIA (or a similar model) would prove highly beneficial to facilitating the integration of tools like PIANOTO into the workflow of musicians, enabling the creation of music directly in synchronization with other musical sources.

ETHICAL IMPLICATIONS

ENERGY CONSUMPTION Currently, the PIA model requires a GPU to run on (the hosted demo runs on an AWS-hosted Nvidia T4 GPU), which, on top of inducing high energy consumption, limits the availability of the technology. An avenue for improvement here is the current research on e.g. efficient Transformers or model miniaturization which could help make the backend more lightweight and ideally enable it to run, in JavaScript, directly within the interface, on the user's machine. Still, traditional, "non-smart" piano-rolls are extremely lightweight software and replacing them on a large scale with such AI-assisted technology could therefore induce a very unwelcome increase in computation and energy consumption. Here, advances in making the backend more efficient could actually lead to a detrimental *rebound effect* by making these technologies more widely (over)used.

MUSICAL DIVERSITY The PIA model that the demo uses has been trained on the GIANTMIDI-PIANO dataset [67], a dataset of classical piano comprised (mostly) of western compositions, it is therefore naturally biased towards that genre of music. Two things can be said here:

1. Even then, thanks to the ability for context adaptation exhibited by the model, PIA manages to adapt to other styles of music, unseen in the training set! This is an advantage of the inpainting-based approach, where the conditioning data enables on-the-fly domain adaptation at inference time.

2. The PIANOTO interface is model-agnostic and is open to be used with models trained on other types of music: we encourage other researchers to build inpainting models on other genres of music and use them with PIANOTO!

A further current limitation of the interface is that it adheres to the (western) 12-Tone scale, since the PIA model relies on this scale. Nevertheless, PIANOTO is not inherently bound to this scale: since the interactive aspect of PIANOTO is based only on selecting time-ranges and never explicitly setting specific notes, the interface could potentially be used in conjunction with AI models that support micro-tuning and microtonality. Even more so, we believe that the interface, equipped with a proper inpainting model, could prove very convenient for creating microtonal music with ease! Here, the `Tune.js` library by Taylor and Bernstein [106], a cousin of `Tone.js` designed for microtonal music, could prove useful.

5.6 ACKNOWLEDGEMENTS

I personally thank Martin Leblancs, Matthieu Queru and Sandro Sgro for their help on building a preliminary version of a web interface for the PIA model during a development project under my supervision hosted at Sony CSL Paris, as part of their curriculum at French IT School EPITECH. In particular, it is thanks to their review of existing JS-based solutions for representing piano-rolls in the browser that I became aware of the piano-roll implementation in `Magenta.js`. Thanks to this, I could kick-start the implementation of PIANOTO much more easily than if I had had to start designing and implementing a piano-roll from scratch.

Part II

FROM SYMBOLIC INPAINTING TO MUSICAL AUDIO INPAINTING

General introduction and motivation

In the previous part, we have shown how inpainting models of symbolic music enable the design of novel, intuitive interfaces for music creation. We now turn our focus to another aspect of music production: sound design and sound synthesis. In particular, inspired by image inpainting applications, we envision a tool which would enable, via inpainting, to surgically transform timbre aspects of a given sound: boosting the low frequencies, then smoothing out the attack to make it sound more like e.g., an oboe, all via local, context-aware operations. By representing sounds in the time-frequency spectrogram format, such a tool would furthermore have a clear *visual* interpretation of "*painting*" timbres over an existing sound. Can such a tool be built and made to operate in an efficient enough fashion for interactive applications?

Thankfully for this PhD, the answer is *yes*, although it doesn't come for free. Since audio signals are much more high-dimensional than pure symbolic data, Transformer models as used in Part i are intractable on those. Our solution therefore involves an initial detour into the land of quantized latent representations – or "the VQ-VAE" –, in order to compute *compact intermediate representations* of audio signals over(which can be decoded back to audio). By turning the dense, real-valued original representation of audio into compact grids of abstract tokens, this magical ingredient brings us back to the symbolic case studied in Part i finally allowing us to apply similar techniques.

This second part or the thesis therefore opens up in Chapter 6 with a short background chapter introducing Vector-Quantization and the VQ-VAE model. We then introduce NOTONO in Chapter 7. It is both a novel architecture developed as part of this PhD and a web interface for this model which enables users to generate and transform instrument sounds by visual inpainting. NOTONO has been used in real music production settings by multiple musicians as part of collaborations with Sony CSL and we provide an initial user survey on their reception of this tool.

6 BACKGROUND

Learning compact representations of dense signals with the VQ-VAE

In this second part of the thesis, we seek to apply similar ideas of inpainting-assisted music creation to the audio domain, in order to propose novel techniques for sound-design and musically-oriented sound-synthesis and to offer new, more intuitive ways of controlling deep generative models of audio, since these models have somewhat been lagging behind e.g. image generation when it comes to controllability and user interface design, as discussed in Section 3.3.1.

Generative modeling of audio has seen a surge in fidelity in recent years, thanks to advances in neural network-based architectures [3, 35, 38, 84, 109] and more recently, large-scale models such as JUKEBOX [28] or MUSICLM [1]. These new models are progressively bridging the gap with dedicated synthesis software in terms of the versatility of sounds produced without requiring the involved domain-specific knowledge of audio synthesis. Yet these approaches still fall short of providing convenient and interpretable control over the texture and structure of the generated sounds.

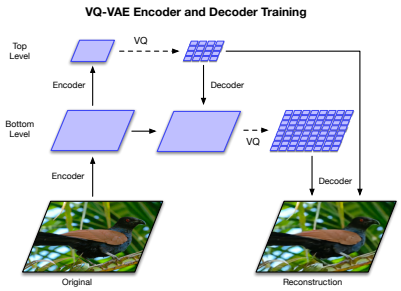
Motivated by the results obtained in the first part of this thesis on symbolic music, we propose to approach the problem of controlled audio generation through the same angle of devising appropriate, *human-compatible scales of interaction*.

Naturally, we might be keen to directly apply similar ideas to the ones developed in the first part, and train e.g. inpainting-enabled Transformer network on audio datasets to then enable inpainting-based transformation. Sadly, this would not work at all as is, since Transformer models such as the ones used in the PIA model presented in Section 5.2 are typically computationally very heavy and would therefore be highly intractable on high-dimensional data such as high-definition images or audio. To circumvent this issue and benefit from the high modeling capacity of these models, the VQ-VAE model, initially proposed for the generation of images [89, 108], offers a powerful

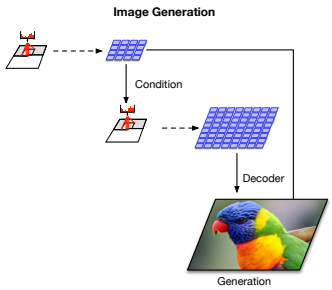
solution. Indeed, this general framework allows one to effectively convert dense data to a *downsampled compact, symbolic, token-based higher-level representation*.

PROPOSED APPLICATION In the context of the NOTONO model presented in the next chapter, it is this intermediate representation which we *then* efficiently model using powerful Transformer-based models for interactive transformation and generation. Additionally, by carefully selecting the downsampling ratios involved in the VQ-VAE model yielding this higher-level representation, we can design the resulting downsampled scale (that is, a 2-D slicing of the original time-frequency representation of the sound into different, larger zones of a given duration and frequency range) to be conveniently manipulated, including on smaller, touch-screen-based mobile devices.

6.1 VECTOR-QUANTIZATION AND THE VQ-VAE



(a) Overview of the architecture of a hierarchical VQ-VAE. The encoders and decoders consist of deep neural networks. Here, the input to the model is a 256×256 image that is compressed to quantized latent maps of size 64×64 and 32×32 for the *bottom* and *top* levels, respectively. The decoder reconstructs the image from the two latent maps.



(b) Multi-stage image generation. The top-level prior model (implemented here with a PixelCNN) is conditioned on the class label, the bottom level prior (a PixelCNN too) is conditioned on the class label as well as the first level code. (The example image with a parrot is generated with this model.)

Figure 6.1: VQ-VAE-2 architecture (Picture taken from [89]).

To this end, the VQ-VAE defines an auto-encoding architecture using a hierarchy of *discrete* (integer valued) latent variables. More precisely, a VAE's latent space is made discrete through *vector-quantization*, a classic dictionary-learning algorithm [15]: the latent representation computed by the VAE's encoder is first sliced into small *patches*, as shown on Figure 6.1a with the sliced representation of the latent

representations, where each smaller square corresponds to one patch. Then, each of these patches is projected onto a learned *codebook*, that is, a fixed-size dictionary of so-called *code vectors*. This projection effectively replaces the latent representation with a *codemap*, i.e. a discrete grid of codebook indexes, where each index corresponds to a sub-patch of the original representation and links to the element of the codebook which is closest to the original value of the patch, hence the *quantized* denomination. The decoder in turn receives only this projection of the encoder's output onto the dictionary, i.e. the dense representation obtained by retrieving for each codeword index at each sub-patch position the codeword at that index in the codebook and inserting it for the patch. By jointly training the encoder-decoder architecture to minimize reconstruction error *and* the dictionary to best approximate the outputs of the encoder, the model is expected to learn a valuable dictionary, able to approximate as diverse inputs as possible given the limited number of codewords we allow it to handle.

At its very core Vector-Quantization (VQ) can be introduced in any neural architecture as a simple layer with associated trainable codebook $\mathbf{e}_k, k \in 1 \dots K$ using the following non-linear activation:

$$\text{Quantize}(\mathbf{x}) := \mathbf{e}_k \text{ where } k = \arg\min_j \|\mathbf{x} - \mathbf{e}_j\|$$

The gradients of this operator are then computed during back-propagation using the so-called *straight-through estimator*, a common trick for back-propagating through non-differentiable functions, that is, the back-propagated gradients flow through the discrete arg min operation as if it were the identity function. This can be implemented in common deep learning frameworks such as PyTorch or TensorFlow using the *stop-gradient operator* $\mathtt{sg}$, leading to the following implementation:

$$\text{Quantize}(\mathbf{x}) := \mathtt{sg}\,[\mathbf{e}_k - \mathbf{x}] + \mathbf{x}$$

After learning a VQ-VAE, one can turn the problem of generating continuous-valued data into a much more tractable discrete sequence generation task over a typically small dictionary (in the order of a few hundreds of codewords). This is done by converting all of the available training data to the discrete codemap format and training prior models on those compact representations instead of training them on the much larger original data. Then, at inference time, one can generate new data by first sampling codemaps from these learnt codemap prior *then* converting these sampled codemaps to the original

data format (images, waveforms, spectrograms...) using the VQ-VAE's decoder, as shown of Figure 6.1b.

6.2 HIERARCHICAL ENCODINGS WITH THE VQ-VAE-2

Furthermore, in the hierarchical approach proposed by the VQ-VAE-2 [89], an extension to the original VQ-VAE model, multiple such discrete latent spaces are learned at varying scales, as shown on the left-hand side of Figure 6.1a, allowing each of these abstract representations to capture specific aspects of the input data, at multiple scales. In the two-layers case, a small *top* latent codemap is obtained from a first – larger – *bottom* codemap by further downsampling it and then, again, vector-quantizing it. In this setting, the highly downsampled top codes can be interpreted as depicting the overall, large-scale structure of the images, whilst the bottom codemap decoder, which receives this global structure as conditioning, is tasked with refining it with finer-grained detail. This conditional dependency of the decoders in higher-resolution layers on the layers above them is shown on Figure 6.1a via the arrow going from the top codemap to the bottom codemap.

6.3 EXISTING APPLICATIONS TO AUDIO

We note that VQ had previously also been applied to the generation of audio signals, for example audio modeling it is presented as an example application in the paper on the VQ-VAE[108] and, in a slightly modified version, in a follow-up paper focused on large-scale audio generation [29]. It was also applied to the generation of instruments sounds (with a focus on timbre transfer applications rather than pure, open-ended sound-design as we seek to achieve) in a more recent paper on learning timbre representations [9]. In all cases, it is validated as a sound and effective approach to learning intermediate representations. Of particular interest is a work that was conducted concurrently to ours on NOTONO and gathered much public interest upon its release in 2020: the JUKEBOX model by Open AI [28]. JUKEBOX shares a very similar approach to ours of combining VQ-VAEs with Transformers, albeit with very different goals. Indeed, VQ is used

in JUKEBOX as an information bottleneck allowing to scale *very* large Transformer-based autoregressive models of mixed, polyphonic audio to large temporal scales of the order of several minutes. Controlling the models proposed in these papers is therefore highly challenging, if only due to the high computational load they incur: generating a minute long of audio takes 8 hours with the highest resolution prior model, making this approach orthogonal to our goal of interactive control. Our approach, whilst much more modest in its scale since it only targets the simpler but significantly more tractable task of single instrument sound generation, is nevertheless designed *for* interactivity and reaches real-time generation speeds, making it compatible with interactive usage for sound design.

6.4 CONCLUSION

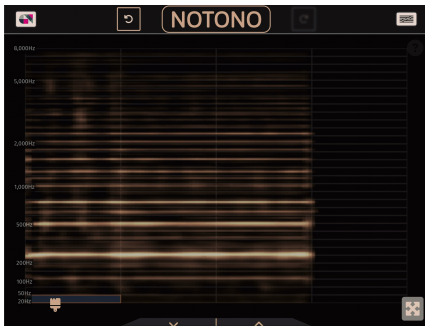
The VQ-VAE approach has been pervasive in recent years in neural modeling, thanks to the ability it provides to rather easily scale models to previously unattainable performances. Equipped with these concepts, we propose to apply it in the next chapter specifically to the design of a novel interactive modality for sound design, making full use of the structure and scale of the higher-level representations it allows us to construct.

7 | NOTONO

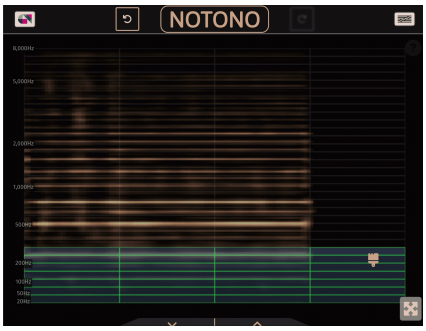
Spectrogram Inpainting for Interactive Generation of Instrument Sounds

♪ NOTONO, NOTONO ♫

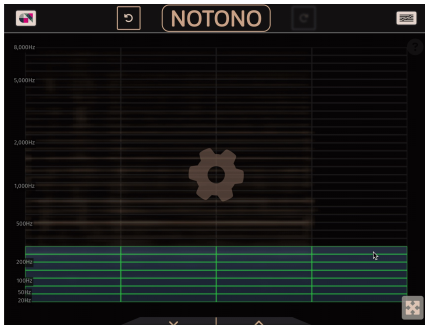
Theme to My Neighbor Notono
Joe Hisaishi



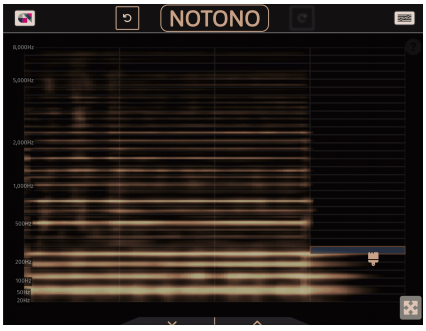
(a) Initial flute sound, generated by NOTONO. The user's *Brush* is located on the lower frequencies, at the beginning of the sound.



(b) Click-and-drag to select a zone with the *Brush* tool. Release the pointer to trigger an inpainting of the selected zone, in green.



(c) A loading animation is displayed and the interface is locked for inpainting interaction during computation.



(d) The displayed spectrogram is updated with the transformed sound. Repeat the operation as desired!

Figure 7.1: Using NOTONO to augment the bass frequencies of a flute sound with subs from an organ, all in just one click.

Rationale

Modern approaches to sound synthesis using deep neural networks are hard to control, especially when fine-grained conditioning information is not available, hindering their adoption by musicians. Conversely, by allowing users to selectively regenerate local parts of an input media conditioned on the surrounding information, inpainting techniques offer a promising approach to the interactive use of machine learning.

I investigate the potential of the inpainting paradigm for assisted, interactive sound design. To this end, we both devise a novel deep learning architecture for the inpainting of instrument sounds in time-frequency representation, and develop a web interface, called NOTONO, to interact with this model in professional music production contexts, allowing to regenerate parts of the time-frequency representation of a sound in a visual, paint-like fashion.

In order to make use of powerful, but computationally demanding, symbolic sequence inpainting models – specifically, Transformers –, I adapt the VQ-VAE-2 image generation architecture to spectrograms. This allows us to convert dense, real-valued spectrograms into compact discrete codemaps on which it becomes computationally tractable to train and run inpainting-ready Transformers. I trained this architecture on the NSynth dataset of instrument sounds [34].

I validated the proposed approach through extensive usage sessions in real-world music production contexts with multiple, professional musicians. This longitudinal study was doubled with a lightweight user survey I administered. The results of this user study are provided in the next chapter.

NOTE This chapter is an edited and updated version of a paper that was published at the 2020 Joint Conference on AI Music Creativity. The paper can be found here: https://arxiv.org/abs/2104.07519.

As for the previous chapters, I invite the reader to watch demo videos of NOTONO to grasp the ideas before delving into the details. First is a very short video[1] describing the general idea of NOTONO (note that the piano sequence driving this demo was actually generated by the PIA model!). Second[2] is a more in-depth demo showcasing how one can shape several sounds for various types of instruments with NOTONO, in the context of the creation of a simple song.

1 https://youtu.be/SzlxPTIcxnY
2 https://youtu.be/B81zeInaHZc

7.1 INTRODUCTION

Given the typically high sampling frequency of audio signals, applying inpainting models to those straightaway is intractable. We thus take clue from recent work in image modeling and propose to tackle the problem of controlled sound generation through a two-step approach. First, a VQ-VAE-2 encoder-decoder network is trained, via a standard reconstruction task, to compute a highly compressed representation of sounds through dimensionality-reduction and discretization, with efficient encoding and decoding. The obtained representation can be seen as an abstract, positional map of indexes of medium-scale time-frequency atoms (between half a second and a second of duration and around 200Hz of bandwidth, on a Mel-like logarithmic scale) which, when combined on the time-frequency plane, recreate the original sound. It then becomes possible to train powerful Transformer-based sequence generation models on these compact maps for interactive generation by inpainting. Through training on a corpus of sounds (in our case the NSynth dataset of instrument sounds, comprised of a diversity of instruments sampled across five octaves and at multiple velocities), the Transformers learn the typical time-frequency structure of natural instrument sounds, and are able to propose reasonable completions for partially masked sounds, which can then be converted back to audio signal.

In order to properly assess the value of this approach in creative context, I propose the NOTONO web interface. The interface provides a direct counterpart to this model, by allowing the user to precisely select areas to regenerate in this compact representation, assisted by the spectrogram Transformers. This open-source interface is meant as a valuable resource for the audio machine-learning community, by allowing other practitioners in the field to plug-in their own audio inpainting models and readily use them in an interactive fashion. We hope it will therefore be of value for scientific dissemination. Nevertheless, this interface also serves a dual purpose as a ready-to-use AI-assisted sound synthesizer for musicians to use in their own creative practice. Preliminary, informal evaluation of this tool in professional music production contexts was performed, and provides material for a critique of the proposed approach in a final section.

Building upon the concepts presented in Chapter 6, I present in Section 7.2 the proposed approach, also providing details on the exact architectures and training procedure used. In Section 7.3 I detail the

work performed on the NOTONO interface and highlight some novel sound transformations made possible by this interactive interface. In Section 7.4 I return to the proposed approach and put it in perspective after extensive usage sessions with musicians, highlighting critical avenues for improvements. Finally, I conclude with a short summary of the contributions and some perspectives on this third and final prototype.

The interface and interactive modality for the resulting NOTONO prototype is presented on Figure 7.1. Clear visual cues are provided to help the user interpret the behavior of the interface and the affordances it offers. In 7.1a, cells highlight on hovering with the Brush tool to signal interactivity. In 7.1b, the user selects a zone for re-generation via click-and-drag. Cells turn to green to signal they are ready for triggering an inpainting operation. On releasing the pointer on 4.1c, the inpainting is triggered and a load screen is displayed to signal the ongoing computation. Finally, on 4.1d, the resulting, inpainted sound is received and the spectrogram is smoothly updated.

7.2 A VQ-VAE ARCHITECTURE FOR INSTRUMENT SOUND INPAINTING

We start by constructing a compact representation computed by a VQ-VAE-2 based on a spectrogram-like time-frequency representation taken from the GANSYNTH paper [35]. Then, this representation can be used as input to Transformers, powerful sequence modeling architectures. The VQ-VAE-2 therefore focuses on learning to generate local timbre using a limited dictionary of time-frequency atoms and provide a localized, structured map of these atoms allowing to reconstruct any given sound by feeding such a map through the model's decoder. This decoder operates by retrieving, from its dictionary, the elementary time-frequency patches at each position of the input map, thus reconstructing a very low-resolution version of the target sound, which is then reconstructed back to a full version by the (convolutional) decoder network. The "Spectrogram Transformers", in turn, operate solely on these abstract maps and learn the natural distributions of such structural descriptions. They indeed have no direct knowledge of actual timbre since they are never given an actual sound to look at, they only know of the relative positioning of different abstract timbral elements, and are subsequently able to generate "well structured"

maps. These maps can finally be decoded back to real sounds by the VQ-VAE's decoder.

7.2.1 Compact spectrogram-like representation for intuitive control

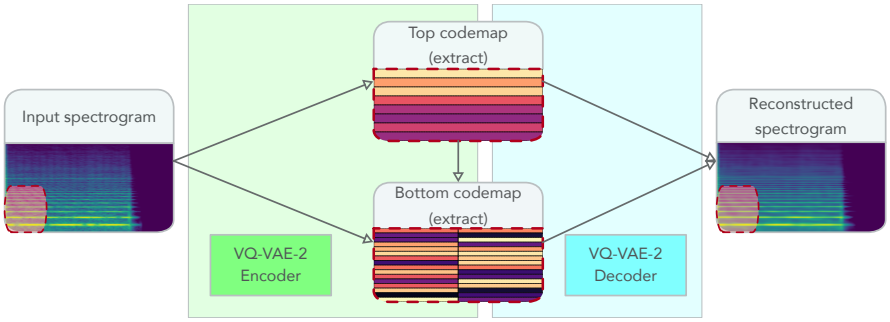


Figure 7.2: Proposed VQ-VAE-2 architecture for spectrograms. The encoder converts 1024×128-large spectrograms into two compact integer maps, c_{top} and c_{bottom} of size 32×4 and 64×8. The decoder is trained to reconstruct the original spectrogram from these discrete codemaps.

With the goal of performing visual inpainting over image-like representations of sounds, we borrow the invertible Hi-Res Mel-Instantaneous Frequency (IF) (Mel-Amplitude and Instantaneous Frequency) spectrogram representation that was shown by Engel et al. [35] to provide the best results for generation with convolutional neural networks. This representation, which performs a temporal unrolling of the phase channel of the spectrograms, introduces smoothness and stability to the phase component of the spectrogram as opposed to directly learning to model the phase. Indeed, directly modeling the phase of spectrograms is challenging, since it exhibits near-random variations and thus diverts a significant part of the modeling capacity of the models used to approximate its distribution. The Hi-Res Mel-IF representation can furthermore be inverted back to the original phase and amplitude spectrogram at a low computational cost, i.e., without requiring phase reconstruction through, e.g., a phase vocoder. Such a representation therefore allows for *efficient* inversion back to audio using just the inverse Fourier transform.

The auto-encoding-based training of the VQ-VAE-2 furthermore requires introducing a reconstruction loss, as opposed to the adversarial training of GANSynth. We propose to use a slightly modified version of the standard L2-loss: we introduce a phase thresholding operation to actively ignore portions of the instantenous frequency

channel which are not relevant for modeling and would otherwise divert the training gradients away from effective learning. Concretely, we ignore IF reconstruction errors (by setting those gradients to zero) in areas of low sound amplitude, where the IF starts becoming highly noisy [35]. This process thus attempts to shift, during training, gradients to areas with significant amplitude, avoiding to waste modeling capacity and training time on mostly silent areas.

Working on spectrograms has the additional advantage of separating time and frequencies in the input representation. This provides an intuitive interpretation of the latent codes learned by the hierarchical VQ-VAE-2 as time-frequency atoms. Editing the resulting 2D codemaps by inpainting can then be seen as performing localized transformations in the time-frequency plane over multiple frequency bands, akin to the multi-band equalizers musicians are familiar with.

7.2.2 Spectrogram Transformers

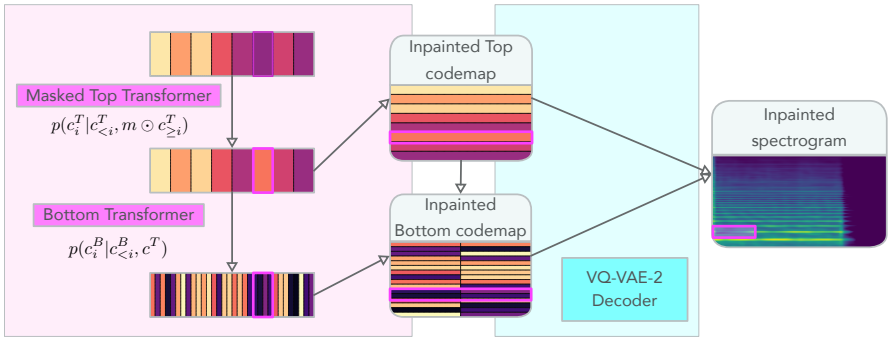


Figure 7.3: Diagram of the proposed spectrogram inpainting process. After a linearization-procedure informed by harmonic structure, sequence-masked Transformers are trained to perform inpainting on the codemaps of the VQ-VAE-2, which allows to sample new sounds by first autoregressively sampling from the factorized distribution $p(c_{top})p(c_{bottom}|c_{top})$ then decoding these sequences.

After training the VQ-VAE, the continuous-valued spectrograms can be replaced with the hierarchical latent codemaps. Generating new sounds then amounts to modeling and sampling from the joint probability $p(c_{top}, c_{bottom})$. We follow the approach of [89] and factorize this probability into:

$$p(c_{top})p(c_{bottom}|c_{top})$$

This is also in line with the conditional structure of the VQ-VAE-2 encoder. We therefore train an autoregressive model for the top codes and another one – conditioned on the top codes – for the bottom codes. To this end we adapt the VQ-CPC architecture [48] for hierarchical discrete latent variables. For the top codemap $c^T_{i\dots N}$, we use a causally masked, self-attentive Transformer, therefore modeling the following, masked probability:

$$p(c^T_i | c^T_{<i}, m \odot c^T_{\geqslant i})$$

where m denotes the randomly sampled *inpainting mask*, used to stochastically hide data during training. Typically, a simple Bernoulli can be used here for each token. This mask then allows, at inference time, to select areas to regenerate, and provide the model with information about the fixed, future tokens, ensuring coherent generation. For the upsampled, bottom codemaps we use another Transformer which can attend via attention to the whole top codemap in a patch-based fashion: indeed, to help the attention procedure, we align patches of the bottom map to the top code which they were upsampled from. We obtain the following formulation:

$$p(c^B | c^T) = \prod_i p(c^B_i | c^B_{<i}, c^T)$$

The resulting architecture is depicted on Figure 7.3.

7.2.3 Dataset, architectures and training

I now present some technical details of the procedure used for training this architecture. I also refer the reader to the following webpage for extensive audio samples of the models: https://sonycslparis.github.io/interactive-spectrogram-inpainting/.

I trained the proposed architecture on the NSynth dataset [34], which is, as described by its authors:

> An audio dataset containing 305,979 musical notes [split into *train*, *valid* and *test*] subsets, each with a unique pitch, timbre, and envelope. For 1,006 instruments from commercial sample libraries, [we] generated four second, monophonic 16kHz audio snippets, referred to as notes, by ranging over every pitch of a standard MIDI piano (21-108) as well as five different velocities (25, 50, 75, 100, 127). The note was held for the first three seconds and allowed to decay for the final second.

As is done in the GANSynth paper [35], we restrict the dataset to sounds with MIDI pitches ranging from 24 to 84, avoiding extreme, unrealistic pitches. Similarly, we also reshuffle the dataset and extract a new 80%/20% train/valid split, since the original train and valid splits are separated along instrument types, which is not meaningful for our application. In the accompanying code, we provide the random seed used to perform this split for reproducibility. We compute the input spectrograms using windows of 2048 samples with $N_{fft} = 2048$ with a 75% overlap factor, resulting in an effective dimension of 1024×128 for the spectrograms. As opposed to the representation used in GANSynth, we lower the Mel-scale break frequency from 700Hz to 240Hz, in order to increase resolution in the low frequencies, bringing more precision in the interactive editing of the lower harmonics.

The VQ-VAE is based on a ResNet architecture with symmetric encoder and decoder. The bottom encoder is composed of 2 residual blocks containing each one 3×3 Conv2D layer and one 1×1 Conv2D layer, followed by a succession of 5 2×2 Conv2D layers with stride 2 for additional dimensionality reduction. It uses ReLU non-linearities. The decoder has the exact same architecture but mirrored, using transposed convolutions. The VQ-VAE-2 has a dictionary size of 512. It is trained for reconstruction of the Mel-IF representation using the phase-masked L2 loss described in 7.2.1, using the ADAM optimizer. After training the VQ-VAE, we use its encoder to extract the latent codemaps for all of the sounds in the training dataset. The bottom and top Transformers can then be independently trained on these codemaps. Here, we train in an autoregressive fashion with the RAdam optimizer [72] and a label-smoothed prediction loss [77]. Both the Top and Bottom Transformer models are encoder-decoder Transformers with the encoders and decoder each containing 6 layers with 8 attention heads.

7.3 PROPOSED INTERFACE

We believe that generative models for music only reveal their potential through actual usage in music production contexts, which requires providing musicians with intuitive interfaces to these complex models. In line with this philosophy, I propose an open-source, TypeScript-based web interface that allows to generate and edit sounds using the

Figure 7.4: Screenshot of the proposed interactive web interface, NOTONO, on desktop and on mobile.

models presented in this project[3], using a PyTorch-based computing back-end. This interface is represented on Figure 7.4.

The *modus operandi* for NOTONO is a simple one, akin to classic image editing software, which we can expect the general audience to be familiar with. An initial sound is either generated from scratch using the autoregressive models, sampled from a dataset or provided by the user through drag and drop. The VQ-VAE-reconstructed spectrogram is displayed on the interface, along with a grid overlay depicting the cells of the VQ-VAE's latent codemaps. The user can then perform inpainting operations as described in Section 7.3.1 by selecting zones on this grid, either on the coarse top codemap or on the more detailed bottom map.

I note that the possibility of using different instrument and pitch constraints for successive inpainting operations opens the way to the generation of heterogeneous spectrograms, unseen in the training dataset – such as inpainting the attack of an organ sound with a `"Guitar"` instrument constraint to obtain a plucked organ. The successive inpainting operations performed in Figure 7.5 are an example of this use case. I finally observe that, due to the inherently linearly-scaling duration of the autoregressive generation, each single token-wise prediction of the spectrogram transformers should be fast if responsive interaction is desired. Currently, typical local operations (e.g. regenerating one second of sound across the whole

3 The NOTONO interface, along with demo videos and instructions for starting a local Docker container running the relevant inference server, is available from https://github.com/SonyCSLParis/music-inpainting-ts. The PyTorch code for the model is available at https://github.com/SonyCSLParis/interactive-spectrogram-inpainting.

frequency range) take around half a second to complete when running the models on GPU, ensuring seamless operation.

The interface supports importing any sound into it via drag'n'drop for analysis by the VQ-VAE and subsequent editing. Thanks to the Web MIDI API, plug-and-play support for MIDI controllers has also been integrated. These MIDI controllers can either be actual MIDI keyboards connected to the user's machine with on-the-fly re-pitching of the generated sound to play it back as a sampler, or virtual MIDI connections originating e.g. from Ableton Live. The latter allows using NOTONO as a proper synthesizer, sequenced within Ableton Live, directly from the user's browser. In this setting, inpainting operations can be performed on the fly, with the sound smoothly transitioning on each inpainting operation via a slight cross-fade.

Furthermore, care has been taken to try and make this interface responsive, that is, ensure that it shall be usable both from a desktop computer and from smartphone. This felt important to ensure wide accessibility, valuable for dissemination purposes. Furthermore, touch-screens lend themselves very aptly to the interactive nature of this tool, and make it very convenient to select, in a single finger movement, the zones to regenerate. By adding a small haptic feedback whenever cells in the inpainting grid are activated or deactivated, I furthermore sought to add a sense of tangibility to these interactions, making the spectrogram of NOTONO a truly malleable sound object.

As a final remark on the choice of building a web-based prototype for this interface, I observe, half jokingly, that this fully remote and installation-free deployment approach proved beneficial during the successive COVID-19-related confinements, when in-person setup of the prototype at the collaborating musicians' places would have been impossible.

7.3.1 Inpainting operations

On Figure 7.5 I display uncurated samples for three possible uses of the model. First, a spectrogram for a 4 seconds organ sound at MIDI pitch 60 generated by our model is shown. Three successive inpainting operations are then presented. The first two examples operate on both codemaps, by first auto-regressively resampling the displayed portion of the top codemap, then resampling the aligned tokens in the bottom codemap. The third one only regenerates the bottom codemap, keeping the top one fixed. In the first example,

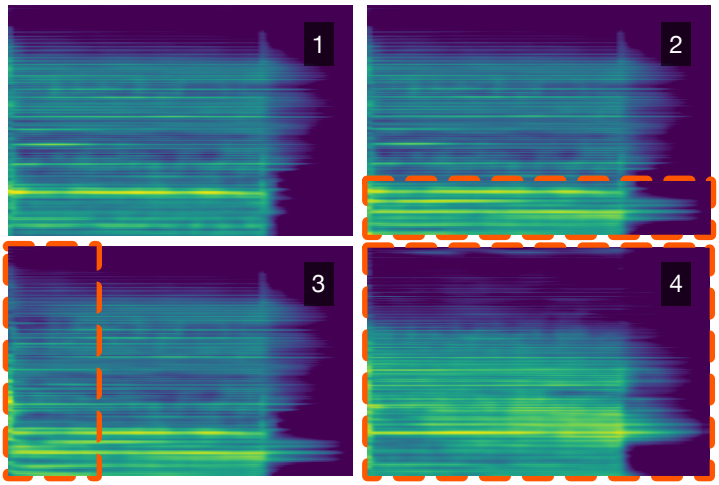


Figure 7.5: Successive inpainting operations over a generated sample. (1) Generated sample ; (2) inpainting of the lower frequencies over the whole duration of the sound ; (3) inpainting of the full first second ; (4) inpainting of the full bottom codemap with the top codemap fixed.

the five lowest frequency bands of the top map are regenerated over the whole duration of the sound with constraint `"Bass + pitch C1"`, resulting in a visible increase in lower-frequency energy. Then, the first second of the spectrogram is regenerated over all frequency bands with constraint `"Mallet + pitch C3"`, resulting in a more marked attack. In the last step, the top codemap is kept fixed but the full bottom codemap is resampled conditioned on it, with a constraint `"Synth Lead + pitch C5"`. We can see that the global structure of the spectrogram remains similar, but the local timbre texture indeed changes. I again refer the reader to the accompanying webpage for more audio examples[4].

7.3.2 Operating on longer sounds

Note that the model, although it was trained on sounds all of duration 4s can be used to analyze and transform sound of arbitrary duration, quantized via padding to the resolution of the VQ-VAE-based analysis/synthesis model. This is due to the purely convolutional nature of the VQ-VAE's encoder and decoder, which therefore are flexible in the time-dimension. Sounds of arbitrary duration can therefore be imported into the interface, and inpainting operations can then be performed in a sliding-window fashion, by operating on windows corresponding to the size of the VQ-VAE and Transformers. Here, I slightly

4 https://sonycslparis.github.io/interactive-spectrogram-inpainting

trick the Transformers into believing they are actually operating on sounds of a structure similar to that seen at training time. I do this by generating fake positional embeddings for the longer sounds, since these the Transformers have never seen time position higher than the maximum duration of the sounds. Practically, I decided to assign only the first second of a sound to the true "first-second" positional embedding, conceptually corresponding to the attack part of the sound and similarly assign the very last second of the sound to the "fourth-second" embedding, corresponding to the release part of the sound. I then split the central part in two halves and label all, one-second slices of the first half with the "second-two" positional embedding and the second half with the "third second" embedding, corresponding to sustain and decay parts of the sound. This is arguably both arbitrary and artificial and likely constrains the structure of the generated sounds. Mechanisms to perform sch extension in a more principled way could be the object of future research. I note that similar ideas have subsequently been explored by colleagues at Sony CSL Paris with similar token-based representations in the context of Generative Adversarial Network (GAN)-based musical audio generation [79].

A proposed application of these inpainting methods is for time-stretching. Here, we take an existing 4 seconds sound and extend its stationary sustain part by duplicating the top and bottom codes for second two of four. This is made possible by the fully convolutional structure of the VQ-VAE, which can therefore operate on spectrograms of arbitrary duration. We then make use of the conditional bottom Transformer to regenerate this slice, resulting in a sound that is indeed a smooth, timbre-aware extension of the original sound and not just the result of classic looping.

7.4 CURRENT LIMITATIONS AND PERSPECTIVES

After having implemented this first, usable version of the whole interactive approach, various avenues for improvement arise. First and foremost, I note that the approach overall lacks a quantitative evaluation framework. Future evaluations should ideally be made in a more strict fashion with systematic reporting and comparison across diverse metrics. Relevant metrics include e.g. the Fréchet Audio Distance [66], which seems to have found traction in the audio machine learning community. More crucially, work should be devoted to devising an

evaluation procedure specifically for the inpainting task. For this, inspiration could be found in the relevant literature on inpainting of natural images. Tasks such as missing-frequencies restoration or random-pattern erasures reconstructions could provide a valuable framework.

I now discuss current limitations over specific aspects of the proposed application, based on personal usage as well as feedback obtained through the aforementioned artistic collaborations, which I explore with more details in the next chapter.

7.4.1 Increasing the audio quality

First, as was mentioned by all of the artists who used the NOTONO interface, the sounds produced by the model all share a set of audible artifacts that should definitely be fixed. These are detailed more, along with audio examples, on the companion webpage, I only list the general failing aspects here. The sounds tend to exhibit amplitude-modulation-like large-scale artifacts, and overall lack variability, with harmonics tending to remain exaggeratedly stationary, to the points that the resulting sounds feel very artificial. These issues could be traced back to several aspects of the current VQ-VAE-2 architecture in place: first, it could be beneficial to increase the dictionary size of the VQ-VAEs, to allow them to learn mode diverse timbre textures. Second, the encoder-decoder architecture currently used is purely deterministic which could be too constraining. Loosening this constraint could open up the way to more non-linearity and non-stationarity for, e.g., better vibrato reconstruction and better noise-components approximation. Third, the architectures themselves used for the encoder and decoder, which are very standard ResNets designed for natural images, could very likely be improved with stronger audio-related inductive biases. For this, inspiration could be taken from the recent works in disentangled audio generation, both in speech [8] and musical sounds [36]. Finally, the models currently generate sound at a sampling frequency of 16kHz, this limit would need to be lifted to at least 44.1kHz if the tool is to be properly distributed as a professional synthesizer.

The challenge in all this is to obtain higher fidelity for the analysis-synthesis model without sacrificing on its efficiency, so as to ensure smooth interaction with the model. Recent advancements in diffusion based models for audio [41, 54, 95] could be of interest here, since these

have proved to be computationally efficient with high reconstruction abilities.

7.4.2 Increasing the Transformers' efficiency

A somewhat easy path for improvement resides in the integration of recent advances in Transformer-based architectures. Indeed, following the line of work on Efficient Transformers [39, 64, 105], which has proved that it is possible, under loose assumptions, to reduce the complexity of the attention mechanism to a linear one, a string of papers have proposed ever-more efficient reformulations of the attention module. Integrating those advances to our approach could, almost for free, significantly reduce the computational load of the inpainting operations. Ultimately, the goal here would be to be able to run the models on device, directly on the users' computer or smartphone, alleviating the need for remote, dedicated GPU-based servers. On top of dramatically reducing the cost of distributing the NOTONO interface to virtually zero. As an additional gain, this would remove the need for sending the users' audio files to a remote server for analysis, alleviating potential privacy issues of handling user-sent audio data.

7.4.3 Increasing the diversity of the produced sounds

As mentioned, the artists who used the interface all complained of the too limited diversity of the sounds that it produces. This can be in part attributed to the limitations in audio fidelity of the models, yet also likely stems from the dataset used for training the models. Indeed, the NSynth dataset is altogether a pretty dull dataset, which certainly has the advantage of being very nicely structured and convenient for the prototyping of machine learning approaches, but where most of the sounds are, per se, musically not very interesting. One area of work will thus be to retrain this architecture on more musically interesting datasets. Training the model with dataset comprised of sounds extracted from a single synthesizer could also prove valuable in that it could allow for interesting audio textural transfer, by offering an analysis-synthesis model able only to reconstruct sounds with timbre elements extracted from the training synthesizer. Finally, for creative purposes, it could be valuable to train this architecture with natural sounds such as car sounds or wind noises, allowing to produce even more diverse instrumental hybrids.

Furthermore, the temporal structure of the sounds can often be very frustrating. Since all sounds in the NSynth dataset are of duration 4 seconds and are held exactly from second 0 to second 3 and left to decay for 1 second, this has rigid consequences on the types of sounds produced by the interface. First, if there is silence on a given frequency-band at second 2, then all inpainting operations o that frequency band on second 3 will inevitably lead to silence, since the Transformers have never seen during training sounds where energy suddenly appears from silence right within a sound. This can be counter-intuitive to the users who would expect inpainting operations to produce substantial transformations on the selected zone. Overall, the temporal structure of the sounds can feel very constrained, partially introducing a temporal envelope to the sounds produced, albeit a very rigid one. As a very rudimentary initial solution to this issue, a mode was added to the interface that allows to perform inpainting operations *without* calling the Transformers. There, codes are just sampled in a uniform, random fashion from the VQ-VAEs codebooks, allowing to insert unexpected timbral textures into the sound. But this "breaks" the model in that it immediately leads to sounds far away from the natural training distribution, making subsequent inpainting operations with the Transformers not well founded. It is is therefore not a satisfactory solution in the long run.

One envisioned possibility here would be to completely remove temporal aspects from the proposed model and instead only train on sounds in stationary regime (i.e. in the "sustain" part of their temporal envelope). In this context, all inpainting operations would always lead to transformations of the sound, since there would likely always be audio energy to manipulate. Such sounds could be made to have arbitrary duration as is typically done in software samplers by looping them with a small cross-fade leading back from the end of the sound to its beginning. Envelopes would then be added externally, e.g. in the sampler module of the user's DAW.

7.5 CONCLUSION

I have presented an approach to interactively generating instrument sounds by inpainting. To this end, I adapted the VQ-VAE-2 image generation model to spectrograms via a representation inspired by the Mel-IF representation from GANSynth. This allows to encode sounds

into compact discrete sequences, greatly compressing the informational complexity of the data. I then introduced efficient hierarchical Transformers with sequence-masking, in order to model a factorization of the joint probability of the top and bottom VQ-VAE-2 codemaps. These models are trained to perform autoregressive prediction, but incorporate information both form the past and the future of the sequences, so that they can be used to perform inpainting on the codemaps at inference time. I introduced a web interface, NOTONO, that allows to use the trained models to generate and edit sounds, which I distribute in an open-source fashion, for researchers and musicians alike.

LIMITATIONS AND PERSPECTIVES The work presented here still represents an initial approach to the problem at hand, and there are many potential avenues for improvements, in particular regarding the architecture used for sound analysis and synthesis, as well as the dataset used for training the model. I hope that by providing the conceptual framework behind NOTONO and an easy-to-use, reactive interface for prototyping models in real conditions, we can spawn some interest from the community in inpainting-based transformation of sounds. In this perspective, it should be noted that similar ideas of processing sound spectrograms via image networks have recently[5] gained traction in the MIR community in the form of the Riffusion model [41]. Riffusion is a version of the Stable Diffusion [94] image generation model, fine-tuned on spectrograms. Although there are some unclear aspects about the training (in particular the training data used for fine-tuning) of this model, one very interesting aspect put forth by the authors is that, by ultimately being the same model as the standard Stable Diffusion architecture, this model can readily be used with any interfaces built for Stable Diffusion, such as the Web UI of AUTOMATIC1111 [118], supporting transformations by inpainting. It could therefore very well be that the wave of enthusiasm for Stable Diffusion-based image generation will in turn bring a new wave of models for audio generation, with inpainting as a core feature. Thanks to it being open-source, NOTONO could be adapted to these new models for exciting interactive usage.

THIS IS ALL NICE, BUT IS IT ANY GOOD? Finally, and moving onto the next chapter, the proposed approach naturally remains to

5 As of December 2022.

be properly evaluated, so as to assess its interest to musicians and potential users. This is the object of the next chapter, wherein I report on multiple collaborations and usage sessions of the NOTONO prototype by professional musicians, resulting in a longitudinal evaluation of this tool and multiple concerts and official music releases involving NOTONO. These usage sessions in real music production contexts have provided us with valuable feedback, allowing us to constantly improve the prototype, as described in this chapter.

8 MUSICAL COLLABORATIONS

An artist centric evaluation

As part of my work within the music team at the Sony CSL Paris lab over the course of the PhD, I had the chance of working in close collaboration with several professional musicians. These collaborations took the form of workshops or residencies, with artists interested in the music creation tools developed by the team initiating long-term partnerships to create "actual" music using the tools developed at the laboratory. The idea, also discussed in a Transactions of the International Society for Music Information Retrieval (TISMIR) journal paper written by colleagues of mine at the Sony CSL laboratory [27], was for these collaborations to be benefiting for both parties, with tacit engagements on both sides. The researchers, developers and domain experts within the CSL music team provide guidance and close support to the musicians, potentially extending to providing custom extensions and features for the tools if needed by the artists and possible according to technical limitations and time constraints. In return for this and in order to anchor the resulting creative practice within the framework of real-world music production, the musicians are expected to aim – as much as possible – for officially releasing (in the form of their choice) the outputs of the collaboration to the public. This served to avoid facing the ecological validity gap which could have emerged had we only focused on testing the tools in a controlled, confined lab environment without the actual goal of releasing music [18, 107]. Indeed, the intent here is to properly evaluate the tools in the actual context for which they are being designed: music production *by musicians* in a studio (or on stage), *not* in a research laboratory. This music-oriented setting furthermore was simply enjoyable, making for a very pleasant experience on the side of the researchers, and I believe, on the side of the musicians too, a key factor for their continued motivation crucial in such long-term collaborations. As Uèle Lamore, who was a resident in 2020 and 2021 summarizes it: "We spoke the same language: we all

wanted to just experiment, make good music, push the tools forward to serve the music, help each other during the process and have fun".

IMPACT ON THE PRESENT WORK As far as this thesis is concerned, these collaborations specifically gave me the opportunity to develop NOTONO, one of the three tools presented in the PhD, with direct and constant feedback from musicians who tested it, some of them using from its very infancy in 2020 and others after much of the initial development phase was finished already, nevertheless still providing me with valuable insights into the potential usage of the tool as well as prompting additional improvements to the design of its UI and UX.

In particular, young French-American composer-arranger-conductor Uèle Lamore and French pop singer-musician-songwriter Whim Therapy (real name Jérémy Bénichou), both close collaborators of the Sony CSL Paris Music Team, were in residency during the early days of my work on NOTONO, creating music using a variety of the tools developed at the lab [27]. UX aspects of NOTONO such as the support for MIDI-In along with the ability to pitch-shift the generated samples on-the-fly for easy evaluation of the generated sounds by playing actual melodies and chords on an external keyboard were motivated by these collaborations and the desire to provide these professional users with as flawless an experience as possible. The underlying intention was to try and ensure that they could evaluate the tool for its *essential specificities* (the interactive modality and the qualities or limitation of the generative model) and not be biased for or against it by mere integration or installation issues.

These collaborations thus made it possible to obtain a *longitudinal evaluation* of the tool in professional music production contexts with several musicians during separate creation residencies performed at the Sony CSL Paris laboratory over the course of 2020, 2021 and 2022. This study was completed with a short survey conducted by me and administered online via Google Forms between end of year 2022 and beginning of 2023.

8.1 USER SURVEY INTRODUCTION

This survey[1] started with recasting *short biographic elements*, serving to contextualize these users in their musical practice and expertise as well as evaluate their usage of the tool in terms of the *period and creative context* during which usage took place and in terms of the *total duration* of their usage of the tool over the period. It then continued with the administration of the Creativity Support Index (CSI) survey as proposed initially by Cherry and Latulipe [19] and which was specifically designed by its authors for "evaluating the ability of a creativity support tool to assist a user engaged in creative work". It then concluded with a subjective evaluation of the tool through 3 Likert-scale based question and *open-ended* questions. These specific questions are displayed on Figure 8.1.

LACK OF INTEPRETABILITY OF THE CSI Arguably, the results obtained through the administration of the CSI as I have administered it provide much less directly usable insights than the answers to the open-ended questions, as they lack any form of *anchoring* which would result from administering this survey several times with respect either to *other tools* providing similar use-cases or with respect to *other versions of the same tool* with *different design strategies*. I therefore take the liberty of not reporting on those specific results in the thesis. As discussed in Remy et al. [90], there is actually an intrinsic difference between what is defined as Creativity *Support* Tools and Co-Creative Tools, the latter offering to actually (co-)create material instead of "only" enhancing the experience of the user-creator. In our case, it seems clear that the tools presented fall more into this second category, and the questions which constitute the CSI arguably do not directly address the specificities of co-creative tools. Indeed, they focus more on technical satisfaction of the user, rather than on the relationship to the outputs and artifacts provided by the tool. For this reason, I have formulated and included some questions in my short survey that try to *directly* address those points.

CUSTOM SURVEY CUSTOMS Taking inspiration from seminal and visionary NIME papers on music interface design by Cook [23] and by Wessel and Wright [115]. In particular, taking cues from Cook's

1 A copy of which is visible here: https://docs.google.com/forms/d/e/1FAIpQLSdrvzwBaKDOyjiEOGSSwadckg6VRKdfWlvN4cnrWw7NWRohvQ/viewform?usp=pp_url

principle number 6 of computer music controllers design ("Instant music, subtlety later") and from the "Low floor, high ceiling" principle of Wessel and Wright [115] (and its slightly humorous, updated version "High-ceiling floors" in the author's commentary to this paper in a subsequent re-publication as part of an anthology of NIME papers), I derived the following two agreement rating propositions: *"I could get valuable results with the tool quickly after using it for the first time"* (low-floor, instant music) and *"The more I used the tool, the more valuable the results I managed to get"* (high ceiling). These principles generally state that for a newly proposed interface for music expression to be adopted, it should enable immediate and intuitive usage (much like a piano or drums, which can be used to make some noise by anyone irrespectively of their musical knowledge) but should also be rich and hide enough complexity that it can provide for a pleasant learning curve and enable virtuoso usage (much like a real instrument), maintaining excitement in its users along the way and enticing them to delve deeper into the tool – in the absence of which it would remain a mere toy. These principles can also be restated, in the context of UI and interaction design as the following, from Cooper, Reimann, and Cronin [24]: "One of the eternal conundrums of interaction and interface design is how to address the needs of both beginning users and expert users with a single, coherent interface". NOTONO seems to have succeeded rather nicely on both those fronts, though it could still be improved. It received an average of 8/10 to the first statement (valuable results from the first use), and with an average of 6.8/10 to the second (more valuable results over time). This lower score for the second statement can likely be attributed to the main complaint expressed by the users: the limited resolution of the model in its current form, leading to frustratingly similar-sounding results (to some extent) regardless of the input sounds. This is discussed more in the per-user reports.

Furthermore, to assess whether the goal has been met of making these tools easy to integrate into the pre-existing, usual workflows of musicians, I added the following agreement rating statement: *"It felt natural to integrate the tool within my usual music production practice"*. Here, NOTONO receives a generally very positive rating of 8/10, which could potentially be improved through direct integration into DAWs as a Virtual Studio Technology (VST) or Audio Unit (AU) plugin as mentioned by musician Whim Therapy, and probably also by increasing the model's resolution and quality, thus ensuring that the resulting sounds would be more directly usable in general contexts.

In order to evaluate what is a recurring theme and question of AI art practices, I added two open-ended questions aiming at addressing the *relationship* of the artists to the artifacts they created with the tool. Indeed, it is often discussed whether AI-assisted creative operate in the replacement of the artists, depriving them of their authorship and autonomy. Specifically, I asked them the following two questions: *"How would you describe your relationship to the results obtained with the prototype?"* and *"Do you feel like you could have achieved those results without the tool?"*.

Trying to also address the mobile-aspect of these tools, I asked the artists whether they had made use of the mobile versions, if if yes how ; if no, why. None of them did use them professionally to the potential exception of Aline Gorisse from *YA*, who nevertheless did not provide any additional details as to her usage of this mobile version. CHATON writes that "[he] used [NOTONO on a mobile phone] but only for fun, not professionally". Generally, this can be attributed to convenience and habits: Whim Therapy writes that "[he] doesn't produce music on a smartphone.", and DeLaurentis observes that "[she] doesn't use the mobile version because [she] likes the sound of [her] monitoring [speakers]". This is also in line with the professional, studio-oriented context for their usage of these tools: it likely simply made more sense to use the available desktop versions in conjunction with DAWs.

Finally, I have integrated rather open-ended and generic questions, to let the users suggest unforeseen aspects, whether qualities or defaults, of the tools, in their own words.

In total, N = 5 different users thus provided insights into their usage of NOTONO and one musician additionally made use of the more recent PIANOTO tool, answering the same questionnaire a second time with a focus on this other tool. The outcomes of this survey are provided in the relevant chapters for each of the tools. I now introduce the musicians with whom I directly collaborated during the thesis and the respective projects.

8.2 COLLABORATORS AND RESULTS

On Table 8.1, I recall some biographical and context data for each of the musicians involved in this longitudinal user study. As can be seen, all of them are experts with a long experience in music creation, and some of them have used NOTONO for a long time, ensuring their

OPEN-ENDED QUESTIONS I

CONTEXT:

1. What was the context of usage?
2. What task(s) did you use the tool for?
3. How would you define the tool in your own words?

AGREEMENT RATINGS (from 0 to 10)

1. I could get valuable results with the tool quickly after using it for the first time.
2. The more I used the tool, the more valuable the results I managed to get.
3. It felt natural to integrate the tool within my usual music production practice.

OPEN-ENDED QUESTIONS II (emphasis present in the survey)

SUBJECTIVE EVALUATION:

1. What were some things you felt were *missing* in the tool?
2. What were the things that you *enjoyed* about the tool?
3. What (if any) were some things that felt *surprising or exciting* about the tool?
4. What (if any) were some things that felt *weird or frustrating* about the tool?

ON AI-ASSISTED MUSIC CREATION:

1. How would you describe your relationship to the results obtained with the prototype?
2. Do you feel like you could have achieved those results without the tool?

ON MOBILE-BASED MUSIC MAKING (recalling that the proposed tools were also usable on mobile devices):

1. Did you make use of the mobile version of the tool (either on a smartphone or tablet)? If *yes*, how? If *no*, why?

Figure 8.1: Specific questions asked as part of the user survey submitted to the musicians who used one or more of the tools presented in this PhD. The survey was administered by focusing on a single tool at a time.

feedback is not just superficial. The following artist introductions are interspersed with their answers to the biographical questions from the short survey I administered at the end of the PhD. In presenting these artists, I also recontextualize their usage of the respective tools, highlighting the versatility of those uses, in a wide range of genres and styles and with several technical approaches. This serves as a testament to the flexibility of the proposed approach for NOTONO, which, even though it was trained on an arguably simple and small dataset, has enabled these artists to create sounds that matched their respective style.

8.2.1 Uèle Lamore

One of the first artist with whom the music team at the Sony CSL laboratory in its current form collaborated, Uèle Lamore, is a French-american composer, arranger and conductor. The main outcome of her collaboration is *Heqet's Shadow: Return of Glycon*, a 4-track digital EP released by Sony Music Entertainment sub-label *XXIM Records* in April 2021[2]. This EP is "the soundtrack to an imaginary video game" and integrates several tools developed at Sony CSL across each of the tracks.

In particular, one of the tracks, *Corruption Of The Toad Forest*[3], is built around an audio sample that she imported into NOTONO, which already transformed its texture due to the characteristic sound of the underlying VQ-VAE-2-based analysis-synthesis model, and which she further processed using the interactive, inpainting-enabled brushes of NOTONO. She described her collaboration with the Sony CSL music team in an interview produced by CSL and available on YouTube [130].

LIMITED SCOPE DISCLAIMER Uèle was the first musician to extensively use NOTONO, and was faced with various bugs and limitations of the model. In particular, for most of her sessions using it, there was a somewhat critical bug in the synthesis (decoder) section of the VQ-VAE-2-based analysis/synthesis, leading to improper reconstruction of the decoded sounds. I am sadly not absolutely certain anymore which exactly was the bug, but I believe the culprit was a mistake in the reshaping of the discrete embeddings of the VQ-VAE-2 after those have been linearized for editing with the Transformers. This

2

3 https://youtu.be/7IKuqZE3qDY

Name	Years active in music creation	Digital creative tools	AI-assisted music creation	Tool	Date of usage	Period of usage	Total duration of usage
Uèle Lamore	> 10 years	*Very familiar*	*Familiar*	NOTONO	2020-2021	> 1 month	> 20 hours
Whim Therapy	> 10 years	*Very familiar*	*Familiar*	NOTONO	2020-2021	> 1 month	10-20 hours
Whim Therapy	*id.*	*id.*	*id.*	PIANOTO	2022	> 1 month	10-20 hours
Aline Gorisse (YA)	5-10 years	*Familiar*	*Unfamiliar*	NOTONO	November 2021	1 week – 1 month	< 2 hours
CHATON	> 10 years	*Very familiar*	*Very unfamiliar*	NOTONO	Feb.-March 2022	> 1 month	> 20 hours
DeLaurentis	> 10 years	*Very familiar*	*Familiar*	NOTONO	2022-2023	> 1 month	5-10 hours

Table 8.1: Biographical and context data for the collaborating musicians (in chronological order of the collaborations)

Name	Style	Self-described creative practice	Self-described context of usage
Uèle Lamore	Experimental, pop	"Production, composition and performance."	"A creative and research collab with Sony CSL."
Whim Therapy	Pop	"Composing, arranging, producing, playing instruments."	*Participation in the 2021* AI *Song Contest and 4-track EP production.*
CHATON	Pop, French *chanson*	"I'm a composer / producer of pop music."	"Full composition and production of my album VIOLENT BEAU (released on May, 6th 2022)."
DeLaurentis	Electronic, neo-classical, pop	"I have inspiration for Melodies and atmospheric chords."	"An EP where I re-adapt themes from classical music in an electronic way."
Aline Gorisse	Electro-acoustic, contemporary	"Electroacoustic and instrumental composition, improvised music."	"It was for a concert with my duet, YA. We wanted to use [NOTONO] to improvise but we finally used it to create some sounds that we would trigger during the show."

Table 8.2: Stylistic and contextual data for the collaborating musicians (in chronological order of the collaboration)

wrong reshaping was causing a time-frequency mismatch of the desired time-frequency "atoms" of the VQ-VAE with their actual position, resulting in very unnatural sounds that exhibited noisy (and weird) timbre textures. I eventually fixed this bug, in particular it was fixed when the subsequent collaborating musicians started using NOTONO.

UÈLE LAMORE ON NOTONO Uèle nevertheless was highly enthusiastic about the approach from the very beginning and repeatedly validated the UX concepts underlying it. Within the survey, she describes NOTONO as "a "Photoshop" of music" and additionally said in an interview for French magazine *Mixte Magazine*:

> [NOTONO] lets you import any sound into an interface that work like Photoshop. Then you paint over the spectrum with brushes. For sound design, [this approach] will be very interesting.

This general validation of the core concept is further confirmed in the answers Uèle gave to the survey I conducted in 2022:

> QUESTION – *What were some things you enjoyed about the tool?*
>
> UÈLE LAMORE – I thought it was generally a brilliant idea, and I liked the look of the interface.
>
> QUESTION – *What (if any) were some things that felt surprising or exciting about the tool?*
>
> UÈLE LAMORE – The concept of 'brushes' was great and innovative.

Returning to Uèle Lamore's experience with NOTONO, it is essential to observe that even though she validates the concept "on paper" and enjoyed the use of the interface, the experience was still far from flawless, however. In particular, some bugs in NOTONO at the time of her sessions[4], as well as some more essential and still unresolved limitations in the current form of the model, as discussed in Chapter 7 (low resolution, training data, handling of the phase...) made for a sonically imperfect experience. In a document describing her creative process with the Sony CSL tools, cited as part of the paper by Deruty et al. [27], Uèle writes:

4

> The biggest weakness of Notono at [this] moment [in development], was its extreme treatment of sound. This resulted in the creation of very "phasy", filtered, samples with a very peculiar acoustic quality. However this was absolutely perfect to represent the Corruption of the Forest [song title], an unnatural, evil substance slowly spreading like a disease. [27]

Similarly, she writes in my user study that:

> QUESTION – *What were some things you felt were missing in the tool?*
>
> UÈLE LAMORE – [NOTONO] was missing definition and resolution in the treatment of the audio, the result being that ultimately no matter what audio I inserted all results generated by NOTONO sounded the same.

Of interest here is nevertheless the fact that, by enabling simple import of external, user-provided sounds by drag'n'drop onto the interface, Uèle Lamore was able to actually work around those limitations and, thanks to the UI, shape the sounds to obtain satisfactory results.

Continuing on her general feeling towards the tool and the results obtained with it, she writes and concludes:

> QUESTION – *How would you describe your relationship to the results obtained with the prototype?*
>
> UÈLE LAMORE – Conflictual, as I enjoyed using the prototype, but the problem of resolution and sound treatment was very frustrating.
>
> Q. – *Do you feel like you could have achieved those results without the tool? Why?*
>
> UÈLE LAMORE – I do not think so, because only the tool gave me the idea to search for those results. However, I could have gotten something very similar by stacking plug-ins on the original audio track.

DISCUSSION Overall, I believe this represents a very encouraging evaluation of the general concept and UI/UX design of this tool and of its singularity: "only the tool gave me the idea to search for those results". This is a motivation for pushing this concept forward, improving the audio quality and diversity of the underlying models.

Key changes brought to NOTONO by the collaboration with Uèle were:

- The introduction of a more readable scale for frequencies in so that she could better locate the interactions and operations made on the sound, improving the discoverability of the UI.
- The addition of the possibility to add a small fade-in (declicking) to the generated sounds when playing them back directly from the interface. Indeed, as discussed in Chapter 7, as a result of the low overall temporal resolution of the model, the sounds generated by NOTONO tend to exhibit either very strong or very blurry attacks. Those very strong attacks quickly became painful to Uèle when working in a studio on large monitor speakers, and the introduction of the very slight fade-in made operating the tool much more enjoyable.

Finally, the sound created for the Corruption of the Toad Forest track represents an example for a use-case of NOTONO as a *sound processor* and *experimental sound design tool*, making full use of its analysis and synthesis pipeline. This highlights the value in providing models that enable easy *input and output of user-provided data*.

8.2.2 Whim Therapy

The second artist with whom the music team I had the chance of collaborating is *Whim Therapy*, real name Jérémy Bénichou. This collaboration was an occasion for Jérémy, then otherwise active as part of french pop band *Diva Faunes*, to explore new practices in a more open-ended fashion, allowing himself to touch on the more experimental side of his practice whilst remaining close to his pop sensibility. He used the NOTONO tool `in 2020` to "`create a sample database that [he] then used for creating music`" (and as such was equally on the frontline of its development), and the PIANOTO tool `in 2022` for "`composing, trying to get new ideas`", using both tools for `10 to 20 hours` each.

The main outcome of this collaboration is a 4-track EP released by Sony Music Entertainment, *I tried to make music with AI and this happened*[5]. In particular, the track *Let It Go*, which appears on this EP, was created in a tightly-knit collaborative effort involving Jérémy, several members of the Sony CSL team (Cyran Aouameur, Michael Turbot, Amaury Delort and myself) and several musicians and recording engineers as an entry in the 2021 edition of the AI Song Contest under the

5 https://youtube.com/playlist?list=OLAK5uy_lYBKC2ZS4g2kg3w7Rv8mB0qMN18tgFVBM

team name *"Nous Sommes Whim Therapy"* [82], earning the 2nd place in the Public vote of this Eurovision of AI-assisted music creation. Jérémy describes this song as "a soulful homage to old-school music, fulfilled with postmodern elements." The process for the creation of this song involved multiple workshops around several music creation tools developed at Sony CSL, including NOTONO, as well as custom improvements to the tools as suggested by Jérémy. It is described in an interview of Jérémy available on Sony CSL's YouTube channel[6] [131] as well as in the entry page for his song at the AI Song Contest's website[7] [82].

NOTONO was used throughout the EP to process sounds and create new synth textures. In particular, it can be heard on the *Let I Go* song where it was mainly used to create a diverse range of short, glitch-like digital sounds that were used by Jérémy to add some background texture to the track in order to create a contrasting aspect to the mainly analog approach of this composition. This idea was in line with the desire to approach the composition of *Let It Go* as:

> A genuine collaboration between humans and machines. During early stages, we have paid attention to leave some room for the AI to play its part. It has done so wonderfully, leaving us with a song where analog and digital elements are blended to create a unique vibe.

At the beginning of the creation of this song, Jérémy composed some melodic and rhythmic parts which he recorded in a studio with professional musicians, intentionally focusing on recording very sparse lines, with only few notes and strokes. These lines served as a loose grid, a basis to prime the generative AI models with. The intent was that these sparse initial recordings musicians would simply guide the models towards a rhythmic and melodic feel, but would leave plenty of space for the models to "express themselves", thus avoiding too cluttered a final mix, which could have resulted from adding AI-generated content over already rich fragments. Describing this process in the AI Song Contest entry document, Jérémy writes:

> We started the composition process the classic way: First, we chose a chord progression, a vibe, and a tempo. A first studio session allowed us to record guitar, piano, Rhodes, and toplines. Then we wrote the first three sentences of the song. Up to this point, AI was not involved. This first phase allowed us to *create content that would be used as input to our various conditional models*

6 https://youtu.be/pGtrHuYN7Kw
7 https://www.aisongcontest.com/participants/noussommeswhimtherapy-2021

> *later on*. We also wanted the foundation of this track to be as "empty" as possible to leave some room for the AI to feature in. In a second phase, we started to involve the AI tools. [...] Once we got all these, some heavy editing and processing was done to keep only the best elements.

Jérémy additionally used NOTONO during studio sessions with fellow musician Kadebostany, iteratively creating a sample-bank many synthesizer textures obtained by pushing NOTONO to the extremes. To this end, they created and imported already heavily processed and overdriven synthesizer sounds into NOTONO – arguably pushing it very far away from its training distribution. They praised the tool for its ability to yield extreme sonic textures. This represented a pleasant surprise given the mostly clean-sounding nature of the original training dataset, NSYNTH, and is a further argument for the versatility enabled by offering musicians a simple, usable interface over a model with easy input of user-provided data, as it makes possible a convenient user-driven exploration of the space of sounds offered by the model – and the frontiers thereof.

As part of the survey, Jeremy describes NOTONO with his own words as "Painting and merging sounds with a random twist". He mentions as *enjoyable aspects* the "interface, [and the] randomness". He further observes the following:

> QUESTION – *What (if any) were some things that felt surprising or exciting about the tool?*
>
> WHIM THERAPY – Randomness! Never getting twice the same result.
>
> Q. – *How would you describe your relationship to the results obtained with the prototype?*
>
> WHIM THERAPY – It's an interesting approach to music production by painting sounds, and I loved that the results didn't sound like the sounds I usually craft.

This generally constitutes a *positive reception* of the ideas underlying NOTONO and of the stochastic (in)painting-based sound transformation interface. Jérémy nevertheless notes that the tool would have benefited from "integration as a VST or Audio Unit plugin". Indeed, this would enable a more direct integration into DAWs, letting users add sound effects and processing over the sounds generated by NOTONO. This would constitute valuable future work.

As to the points of failure, Whim Therapy cites, like Uèle Lamore before, the presently limited resolution and capacity of the analysis-

synthesis model as a key limitation, causing most sounds to ultimately exhibit similar textures and artifacts. Although he notes that the results remain valuable and usable in the present form of the tool, underlining their unique idiosyncrasies, this limits the versatility of the tool:

> QUESTION – *What (if any) were some things that felt weird or frustrating about the tool?*
>
> WHIM THERAPY – I ended up having the same sound color most of the time.
>
> Q. – *Do you feel like you could have achieved those results without the tool? Why?*
>
> WHIM THERAPY – No, because it has this unique sound and character which is at the same time a benediction and a curse (*cf.* previous answer).

SPECIFIC ADVANCEMENTS This collaboration with Jérémy highlighted the need for an easy undo/redo feature, much requested by Whim Therapy to avoid irrevocably losing a sound after an inpainting operation. This feature was then developed and integrated into all three of the interfaces. The integration of quickly accessible Undo and Redo buttons turned out to be a key aspect of the fully reactive design of NOTONO as well as NONOTO and PIANOTO, as discussed in previous chapters.

As a partial conclusion, the collaboration with Jeremy is another validation of the concept behind NOTONO, with its interface, its stochasticity and the ability to process any input sound cited as key positive factors. Again, however, the key limitation is to be found in the model's current capacity and quality.

CHATON CHATON, real name Simon Rochon-Cohen, is a French composer, writer, performer and singer of pop and urban music, offering a modern take on French *Chanson*. Between February and March of 2021, CHATON spent a full month at the Sony CSL laboratory to create a full album made using only tools developed with the Sony CSL music team – used within the Ableton Live DAW and allowing himself only to additionally use some simple, classic audio effects such as equalizers, delays and reverbs. The result is an album, *Violent Beau*, that sounds very much like a CHATON album but that incorporates rhythmic elements and timbre co-created with AI.

In particular, NOTONO is the only tonal synthesizer used on the whole album, which CHATON used to create various instruments such

as basses, dub-adjacent digital flutes and general synthesizer sounds. These instrument sounds, although entirely designed with NOTONO, are additionally very coherent with the usual sounds that can be heard on CHATON's other non-AI albums, yet with a unique texture, prompting CHATON to try his best to avoid denaturing them when replaying them. Indeed, in personal communications, in response to me asking him about a (quotable version) of his perspective on the resulting sounds within the context of the album, CHATON wrote:

> I made an entire album using NOTONO as my only melodic and harmonic tool. It was my seventh studio album. This album is now an integral part of my discography, it gives rise to radio broadcasts, has hundreds of thousands of streaming broadcasts and takes me on tour as a classically recorded album would have done. For the stage I sampled the sounds of NOTONO rather than trying to replay them or have them replayed, so as not to distort their essence. That is to say if I care about them. It is a crazy experience to have used this revolutionary tool for a composer/producer. Beyond its great technical qualities, it is the way in which it was designed that upsets the genre. Its sensitivity.

This, I believe represents a very pleasant testimony of the versatility of the general approach, in yet another stylistic context.

I now provide the verbatim for CHATON's answer to the survey, which generally hit very close to the general points raised by the previous two artists, strongly appreciating the concept of NOTONO but regretting the current limitations of the model, suggesting that retraining of it on a dataset of vintage synthesizer could prove valuable.

> QUESTION – *What (if any) were some things that you felt were missing in the tool?*
>
> CHATON – Some training on legendary synthesizers.
>
> Q. – *What (if any) were some things that you enjoyed about the tool?*
>
> CHATON – Endless creativity and singularity of each sound.
>
> Q. – *What (if any) were some things that felt surprising or exciting about the tool?*
>
> CHATON – Painting music, obviously.

Q. – *What (if any) were some things that felt weird or frustrating about the tool?*

CHATON – Limited Training of the system.

Q. – *How would you describe your relationship to the results obtained with the prototype?*

CHATON – Very inspiring and exciting for the future of music production.

Q. – *Do you feel like you could have achieved those results without the tool? Why?*

CHATON – Not at all. Everything was based on that tool in my process.

DELAURENTIS *DeLaurentis* (full name: Cécile Léogé) is French female producer-singer-songwriter who works at the intersection of electronica, neo-classical and experimental electronic music. Her work focuses strongly on her voice and modifications of it. As part of the collaboration with Sony CSl, she recorded an EP of electronic re-interpretations of themes from French classical music, utilizing a variety of AI music tools from Sony CSL and IRCAM alike. This EP is set for release during 2023.

DeLaurentis again gives NOTONOa very warm welcome, citing the "unpredictability" of the approach and the expressivity of the sounds it creates as positive factors, making her "very confident" in the resulting material. In line with the importance of the voice in her creative practice, DeLaurentis very quickly tried to apply NOTONO to the processing and transformation of samples of her own voice. She noted that the results were surprising and pleasing, but regretted that the model could not maintain the textual content, phonemes and vocal identity of these sounds, having not been trained on such data. Accordingly, she cites the adaptation of the tool to the processing of the human voice as a promising area of research.

QUESTION – *What (if any) were some things that you felt were missing in the tool?*

DELAURENTIS – Maybe I would like to try the same tool specially dedicated to voice.

Q. – *What (if any) were some things that you enjoyed about the tool?*

DELAURENTIS – The unpredictable.

Q. – *What (if any) were some things that felt surprising or exciting about the tool?*

DeLaurentis – The unique, very expressive sound.

Q. – *What (if any) were some things that felt weird or frustrating about the tool?*

DeLaurentis – Nothing

Q. – *How would you describe your relationship to the results obtained with the prototype?*

DeLaurentis – Very confident.

Q. – *Do you feel like you could have achieved those results without the tool? Why?*

DeLaurentis – No. I'm proud of the result because [they] sound original and surprising.

YA – ALINE GORISSE AND YOUSSRA KHECHAI Finally, in a small collaboration that took place outside of the Sony CSL context, I collaborated with Aline Gorisse and Youssra Khechai, both composers of instrumental and electroacoustic music with an added practice of free-from improvisation and with diplomas from the *CRR 93*, a French Regional Conservatory of music in Aubervilliers. Frequently working together as a duo under the name *YA*, Aline Gorisse and Youssra Khechai used NOTONO as part of the preparation of a public performance which took place December of 2021. I had met with them and presented them the tool, which prompted their interest in it, leading to a small workshop where I explained them how they could use it. In line with their background in electroacoustic composition, they decided to use NOTONO to process and transform samples they had recorded of acoustic instrument – such as droning, stationary cello sounds, which they transformed into pad-like electroacoustic textures.

Shortly after the performance had taken place, Aline wrote the following: "We finally decided against using NOTONO live due to the added constraints, but we did use it to create some pads which we playback back during the performance and the results were really nice! It's really a perfect tool for pads and textures."

Question – *What (if any) were some things that you felt were missing in the tool?*

Aline Gorisse – This tool for me has a very specific use with a very specific sound. You can't compose only with Notono but it's a tool that is interesting to have among your other tools.

Q. – *What (if any) were some things that you enjoyed about the tool?*

ALINE GORISSE – I could use it very easily and it's kind of fun.

Q. – *What (if any) were some things that felt surprising or exciting about the tool?*

ALINE GORISSE – It feels surprisingly fun and simple for a sound treatment that is not usually that easy to do.

Q. – *What (if any) were some things that felt weird or frustrating about the tool?*

ALINE GORISSE – I sometimes would've loved to go deeper in the parameters to shape the sound exactly as I wanted.

When asked about her relationship to the results offered by NOTONO, Aline writes: "Surprising because you can't control everything you do but it is also very specific so you don't have too big of a surprise. And fun and simple at the same time." This specific formulation represented a very nice suprise for me, in that it perfectly sums up the aim I had with the design of NOTONO. Indeed, in highlighting the intrinsic trade-off in this interface between the *surprising* aspects enabled by the stochastic, generative process on one side, but the *specificity* (that is, the control) offered by limiting, by design and for interaction, this stochasticity to a middle-range time-frequency scale, I feel like Aline exactly captured the essence of the principles driving this PhD.

Furthermore, she notes that this tool is valuable even for very advanced experts of electroacoustic music creation as her. Indeed, she observes that, even though similar hybrid sounds as the ones produced by NOTONO could be obtained (by means of slicing and layering of multiple samples), these results would be much harder to achieve with NOTONO:

Q. – *Do you feel like you could have achieved those results without the tool? Why?*

ALINE GORISSE – Yes, I could have but it would have been way more complicated.

IN DEFENSE OF THE VALUE OF BEAUTIFUL INTERFACES As an aside, I find it interesting to observe that two out the five artists mention *visual* aspects of the tools as key positive aspects. Indeed, Uèle Lamore writes "I thought [NOTONO] was generally a brilliant idea, and I liked the look of the interface.", whilst Whim Therapy comments on the animations for inpainted-notes-removal and generated-note-insertions

in PIANOTO and writes that "The midi notes drawing in front of [him] (in PIANOTO) feels like magic". Those two comments constitute what I believe to be a nice (albeit with limited scope yet) initial validation of the value that there might be for researchers to spend time on carefully crafting the *purely visual aspect* of their research tools and prototypes when those are designed to be tested on real-world users. The end result ended appears to be that these end-users are more likely to be directly engaged and more enclined to work with those tools for long durations inspite of their potential technical limitations. Without this time indeed, it is not absolutely certain that Uèle would have had the motivation to keep working with NOTONO, which would have been a pity since this first collaboration significantly drove the initial development phase of the tool.

Yet, in my perception at least, there is a general attitude in "hard" computer science that looks are a *very* secondary aspect, unworthy of the oh-so valuable time and interest of serious, science researchers ; that function only is of the essence. This is an attitude that, at the very least, I was faced with at some points during my work, and had to disregard, if only because I felt that it simply *made sense* to try and provide a pleasant general experience to potential users – out of respect for them. This potential impact of aesthetics of prototypes on the reception users make of them is discussed in several UX research works, such as the paper by Sauer and Sonderegger [96]. Through a user study with several prototypes of varied fidelity and aesthetic quality, the authors of this paper highlight, in their case, a direct relationship between the perceived aesthetics of a prototype and its perceived, subjective usability during testing sessions, as well as an unconscious tendency for users to extend the overall duration of such usage sessions when using the subjectively more aesthetically pleasing prototypes. These results confirm that visual aesthetics are not just a secondary, irrelevant factor, and might help in stimulating long-running usage sessions, which I consider to be *especially valuable in creative tools built for open-ended usage*. This is however only a very initial take on this subject and would benefit from a formal investigation.

I note furthermore, that this care in the visual aspect of the interfaces developed during the PhD also had immediate, useful implications on decisions directly addressing usability aspects. Indeed, spending time on interface design made me aware of one key aspect in the design of these interfaces which is that – obviously! – of the interactive overlay boxes (which describing the high-level representation

computed by the underlying statistical models). This eventually materialised for NOTONO in the addition of a mechanism to color the outline of those boxes, respectively in orange on hover (i.e. when the mouse is over them) and in green for each of the boxes involved in a zone-selection during interactive selections for inpainting. The goal here was to improve *readability* and *accessibility* of the interface, by providing visually contrasted cues indicating the interactive capacity of these boxes, with the colors additionally informing the user of the current type of interaction they are in, with orange indicating simple inconsequential hovering and green implying that releasing the mouse or touch over the interface would immediately result in triggering an inpainting operation. Accordingly, to allow users to nevertheless cancel such interactions midway, I added the possibility to cancel a selection operation by pressing ESCAPE during an interaction.

8.3 CONCLUSION

Overall, I believe these multiple evaluations of NOTONO hint at the very positive reception musicians of diverse backgrounds have offered these tools, as much as they raise obvious and immediate areas of limitation and clear avenues for improvements. These would naturally, for now, be the object of future work.

Part III

CONCLUSION

9 CONCLUSION

In this final chapter, I propose to reflect on the present manuscript and evaluate the contributions, limitations and perspectives of the present work. First, I recast the initial motivations of re-assessing current concepts of spatio-temporal scales of music representation and the potential mismatch between the intrinsic qualities of those representations for *music visualization* and their, arguably limited on multiple levels, practical usability for *music creation*, as well as their more critically problematic cultural and ethno-musicological limitations. I then recall my conceptual and technical contributions as part of this PhD and discuss to which extent I believe the proposed approach of *applying statistical modeling of music to the construction of higher-level, interactive, representations* offers a fresh and convincing take on those concepts, including a discussion of the reception of these tools by professional music producers. Finally, I highlight various limitations of this work in its present form and suggest a list of potential future works to address those as well as opportunities for extension of the present proposal.

CONTEXT AND MOTIVATIONS I opened this manuscript by exposing the motivations for this work., first in the general introduction in Chapter 1, then with more details through elements of music theory, cultural critique and digital organology in Chapter 2. Through a broad overview of the diversity of cultural and technical practices at play in (modern) music creation, I tried to highlight the importance of the notion of *scale* in those practices: ranging from micro-scale such as micro-timings and micro-deviations of pitch for instrumental or vocal performers, or the micro-scale of sampled and synthesized sounds, to the macro-scales of arrangement and composition for the creation of long, coherent pieces. I presented an overview of existing concepts for (western) music theory and music creation that through the prism of those scales, positing that existing abstractions such as the concepts of notes, scales and rhythm exactly emerged as solutions to facilitate the visualization and transmission, first, of music and second, to make it possible to conveniently manipulate those different scales, indeed, to

create and *write* music – and have arguably been successful in one way or another at this, if one looks at the rich history of music in the Western world. Practically, these abstractions represent approximations or quantization of various physical degrees of freedom of sound: pitch representations abstract away the natural fluctuations in frequency of musical sources, rhythm notations adhering to a tight, conceptual grid, abstract away micro-timing and the potential fluidity of time... Those abstractions however, were arguably reasonable and grounded, since those written representations were meant to subsequently be interpreted by human musicians, able indeed to re-inject those lively micro-deviations into the music.

I then suggested however, basing this review in existing literature on music creation interface design as well as cultural critique and the decolonization of the means of music production [2, 74, 99, 110], that these representation scales entail several crucial limitations. First, and even when one only considers those in the context of Western music, a first creative problem arises from the ubiquitous application of these abstractions in tools for digital music creation, for cultural reasons and due to the way these tools emerged over the 20th century through successive refinement of physical acoustic-then-electronic instruments [33, 102]. Indeed, these electronic devices, contrary to human performers, are designed to re-play those representations with exact precision: what was meant to only be a reference has become a strict target ; alas, the Gaussian has collapsed to a mere Dirac.

Furthermore, it is not even obvious that these representations are very convenient for creation: first, even though they have abstracted away many degrees of variation, they still require significant, implicit knowledge (musical background, music theory, rules of harmony... for proper usage). Second, even though, again, they only provide a very quantized representation of music, they still require a high-precision in the interactions performed, and translate poorly in particular to modern, mobile, touch-based devices [68], as shown e. g. by the absence of any interactive performance representation for manual editing in recent, class-leading mobile DAW Ableton Note.

Finally, these pervasive representations are intrinsically bound to concepts of Western music creation, such as rigid time-grids and the reliance on the 12-tone pitch scale, coarsely limiting the applicability of these tools to the creation of music from non-Western practices [52, 98, 110], in spite of their being global standards [40, 101].

GOING BEYOND WITH STATISTICAL MODELING Starting from the observation that the objects at hand (sounds, frequencies, instruments) *are* intrinsically complex, physical objects with many degrees of freedom, and that this complexity is exponentially increased by the combined involvement of multiple musical sources as part of even single, "simple" pieces, I acknowledged that the design of accessible, controllable yet versatile interfaces for music creation is in itself a highly challenging task, and that existing interfaces probably represent a reasonable compromise in the face of these difficulties. If one however were to want to go beyond those existing standards, I suggested that (generative) statistical modeling, for its ability to learn high-level representations of data, whilst simultaneously capturing fine-grained specificities of the data at hand, might represent a very viable solution.

The field is already a vibrant one and recent advances in modeling quality and capacity have inspired this work and shown that these models are likely ready for usage in professional settings [1, 12, 32, 49, 59]. Through a survey of existing AI-assisted tools for music production, in Chapter 3 I nevertheless suggested that, by focusing mainly on the design of models and not of convenient UIs for those, and, in particular, by not directly addressing the key aspect of the scales of interaction, existing tools in this field have mostly fallen short of providing very convincing user experiences – as can be derived from their still arguably limited adoption, in comparison to text-to-image models such as Stable Diffusion and associated interfaces which have gathered significant public focus. Existing AI-assisted tools for music production indeed either offer only very global control through a single latent vector for exploration control (and potentially, additional global attributes like the type of instrument or some high-level characteristics of the sound), in a way comparable to databases augmented with a certain notion of data entry proximity, such as NSynth, GANSynth, DrumGAN [34, 35, 80], or, conversely, propose models with very dense conditioning which practically displace the difficulties in the tackled creative task to another, itself challenging task. This is the case of Tone Transfer-style models with dense conditioning, most of which require their users to provide them with already rich conditioning input [16, 32, 52]. This rich conditioning however is itself also difficult to manipulate, yet the ability for the users to perform subsequent, fine-grained edits of the final results directly hangs on the ability to edit this input, in the absence of which re-recording the conditioning signal represents the sole option.

PROPOSAL I therefore proposed to *directly* address the question of scale and design models *specifically* for the construction of convenient scales of representation, ready for interaction. This proposal is rooted in recent advances in compact, hierarchical representation learning, in particular the VQ-VAE [89, 108] and VQ-CPC [48] frameworks, which both offer to learn compact, discrete, high-level features over data at intermediate scales, potentially integrating multiple such scales as in the VQ-VAE-2 approach. These models have been applied for the generation of music with great results, but never – except for the PIA model and its initial Max4Live integration into Ableton Live, for which PIANOTO represents a direct evolution – under the specific angle of interaction and interface design, mostly as means of training more powerful models, such as in the case of the impressive but hardly controllable JUKEBOX model [28].

However, I posited that these approaches specifically enable the design of radically novel and highly-usable yet *natural* interactive representations, in the sense that the resulting new interfaces *directly* extend on existing visualizations in the form of simple *visual and interactive overlays* over the initial representation. I suggested that this provides what I believe might represent a "best-of-both-worlds" situation: keeping on one side with the valuable *visual* abstractions provided by existing, conceptually quantized, music representations, therefore promoting the discoverability and affordance of these interfaces, but allowing at the same time to easily go beyond the limited resolution of these representations via an auxiliary, interactive generative model, able to perform both global and local edits. Those additional, auxiliary models are trained to conditionally and interactively create music in those intermediate, high-level representations thus exposing a human-usable scale of interaction. Thanks to the initial representation learning step however, the resulting high-level representations can be decoded back into fine-grained representation, re-injecting precise and expressive music details.

CONTRIBUTIONS To illustrate and evaluate the potential of this idea, I designed, implemented and deployed three (web) prototypes, each focusing on a different, standard representation of musical objects at different scales, highlighting the versatility of those concepts. NONOTO, described in Chapter 4, operates on (polyphonic) sheet music ; PIANOTO, presented in Chapter 5, on expressive piano performances in the MIDI format and NOTONO, introduced in Chapter 7, on instrument and

synthesizer sounds in spectrogram representation. All three of these interfaces derive from the exact same principles and furthermore share most of their codebase, showcasing the conceptual simplicity of the resulting design approach, which can be simply summarized as "add clickable boxes over that existing representation". Additionally, the VQ-VAE-2-based generative model supporting the NOTONO prototype was conceived with the help of my advisor Dr. Gaëtan Hadjeres and I implemented and trained it as part of this PhD. All of these interfaces and the associated (trained) models are furthermore freely available and open-source, in what will hopefully benefit both the research community and AI art and music practitioners

The choice of the web platform for the development of these interfaces was motivated first by the appeal of obtaining widely available, installation-free tools, valuable for scientific dissemination purposes and re-usability. Nevertheless, over the course of the PhD, I came to the additional realization that what had initially been developed as exclusively desktop-based interfaces might also be used on mobile phones, thanks to the cross-platform foundations of the web. This was however not a given, due to the challenging nature of mobile DAW design as discussed in the motivations, and it was not obvious from the start whether the resulting interfaces would be efficiently usable on small screens and not much too visually cluttered. The fact however that, after some work on making the interfaces responsive, I had actually managed without too much effort to transfer these interfaces to mobiles devices *without loss of capacities* was the crucial realization for me of the unique potential of the framework described in this thesis.

Furthermore, in order for these prototypes to be directly usable by musicians within their usual music production environment, allowing to evaluate the usability and effectiveness of these prototypes, I integrated multiple standard interconnection modalities into them, some of those representing technical first in the web domain. Support for MIDI-In and MIDI-Out was added everywhere it was possible, as well as the ability for users to easily import their own data for subsequent editing where possible, ensuring that these tools could be used (and, therefore, tested) on musical content relevant to their own aesthetic practices, potentially diverting from the original training distribution of the underlying statistical models, therefore allowing us to assess the versatility of the proposed, machine-learning-based approach. In particular, modern AI models of music analysis enable new

and convenient such workflows, and I integrated direct support for importing *audio* files of existing piano performances into the piano-roll-based PIANOTO interface, through the ONSETS AND FRAMES model by Hawthorne et al. [55] and in particular the JavaScript implementation thereof as part of the `magenta.js` library [126]. Finally, and in what I believe constitutes a first, I implemented support (most conveniently accessed when using the prototype as an Electron application) for the distributed musical synchronization protocol Ableton Link within the NONOTO polyphonic sequencer, which allows to easily synchronize it with compatible external DAWs such as Ableton Live.

9.1 EVALUATION

The resulting musical collaborations have led to multiple commercial releases of music making use of NOTONO for sound design, including a full album by French musician CHATON wherein NOTONO was used to create all instrument sounds (excluding drum sounds). The results of these collaborations and of a small user study I conducted with these expert users are presented in Chapter 8. Generally, the reception of the ideas of an interactive, AI-driven overlay has been very well received, and, in particular in the case of NOTONO, the visual approach to generative sound design it offers was described as "brilliant" and likely to be "very interesting for sound design in the future". Furthermore, all artists were able to obtain results that fitted their respective aesthetics and appreciated the ability to import and transform their own sounds easily. Nevertheless, highlighting a key limitation of the current version of NOTONO, all artists regretted the currently limited capacity and audio resolution and overall quality of NOTONO, agreeing also that training of the model on other datasets, such as ones composed of sounds of vintage synthesizers, could prove very beneficial. Additionally, one user, Whim Therapy, noted that an integration as a VST or AU plugin for direct embedding into DAWs would be desirable in the future.

I believe that the combination of these *in situ* demonstrations and discussions with diverse audiences first and of long-term usage by professional artists second, constitutes a wide-ranging validation of the proposed approach, albeit not a systematic one in the sense of a large-scale user study, which would be valuable future work. The

limitations raised by the artists additionally provide clear-cut avenues for improvements and natural directions for future works.

9.2 LIMITATIONS AND FUTURE WORKS

The work performed here is very much only an initial foray into the design of such interfaces and many directions remain to be explored.

MORE EFFICIENT MODELS FOR WIDER AVAILABILITY In particular, I regret that the ML models underlying the three prototypes described in the thesis all still require being run on a GPU. This coarsely limits the availability of these tools, since only few people have access to deep learning (in particular NVIDIA Compute Unified Device Architecture (CUDA)) enabled hardware. Additionally, this makes for high online deployment costs, since GPU-enabled cloud instances are significantly more expensive than CPU-only ones. I explored these topics during the PhD, in particular through the work on pruning for ML model miniaturization I performed as part of the PhD in collaboration with Dr. Philippe Esling's team at IRCAM [37]. The inspiring example of the Bach Doodle by Huang et al. [57] lets me believe that at least DeepBach might be implemented as a fully client-side model (either in `tensorflow.js` or `onnx.js`), enabling the (cheap) wide deployment of nonoto as a fully static website, without need for server-side computation capabilities. Additionally, with recent progress on making Transformers lightweight and compute-efficient, as discussed in a recent survey by researchers at Google [105], I believe far from unreasonable to hope for a client-side version of both pianoto and notono. This would make these tools widely more accessible, and represents a very valuable avenue for future work.

MORE SYSTEMATIC UX STUDY Additionally, although an initial evaluation has been achieved through the collaboration with musicians, more systematic evaluation of the UX suggestions made in this work is lacking. This could be achieved for instance through ablation studies of key UI elements, as performed by Louie et al. [73], systematic user studies and A/B testing. In this perspective, the web-based nature of the prototypes would represent a valuable advantage for the deployment of such user studies. Here, managing to have the models run on-device in the client-side as mentioned in the previous paragraph

would also be highly valuable to make such large-scale evaluations cost-effective and manageable.

FIXING THE TIMING ISSUE IN PIANOTO This one is pretty self-explanatory (as discussed in Chapter 5).

IMPROVING THE MODEL FOR NOTONO As mentioned repeatedly by the musicians who used NOTONO, its sound quality would significantly benefit from more work on the model. Potential solutions were discussed in Section 7.4. Generally, these include re-training the model on other types of sounds such as vintage synthesizer, scaling the analysis/synthesis model whilst maintaining its efficiency for interactive usage (again, efficient Transformers would help), integrating perceptual losses such as in the RAVE [16] model by IRCAM colleague and now doctor Antoine Caillon or in the works of Sony CSL colleague [81]. Finally, exploring different types of generative models, such as diffusion-based models [41] could prove successful. Very recent, highly impressive advances in high-quality music generation with MUSICLM [1] and SINGSONG [32] – which are also rooted in notions of quantized, hierarchical representation learning, albeit at much larger scales than NOTONO – could also provide inspiration for this endeavor.

EXPLORING THE INTEGRATION OF ADDITIONAL CONDITIONING MECHANISMS The approach proposed in this work is very generic and flexible, and can be combined with multiple additional control modalities, as shown in the case of Fermatas and Chord symbols in NONOTO or with the label-based pitch and instrument constraints of NOTONO. Exploring other combinations of control, either fine-grained or very coarse and global, with a core editing feature centred around inpainting as done here could provided many avenues for novel interfaces and user experiences. Adding support in PIA and PIANOTO (mentioned in the perspectives in Chapter 5) for automatically generating material synchronized with other, user-provided musical sources – provided as MIDI or as audio, or even as a simple beats and downbeats signal – represents an example of such an extension, with dense conditioning information.

EXPANDING INTO NON-WESTERN MODELS AND PRACTICES Following in the lines of Silpayamanant [101], Lumumba-Kasongo [74] and Faber [40], I mentioned in the motivations that a key factor limita-

tion of current music creation interface lies in their strict adherence to Western cultural norms, partially attributable to the challenge of designing interfaces for music creation with finer scales ; such as the tonal scales with more than 12 tones present in classical Indian music or micro-timings in traditional music. I discussed in paragraph 5.5 how I believe PIANOTO for instance could prove beneficial in this context, by enabling to generate music in arbitrary scales without pain, since this interface only relies on the selection of temporal slices of the created music – and already enables to create convincing, expressive performances with fluid timing and velocity as it is. Similarly, NOTONO-like interfaces, by enabling to create sounds directly on a time-frequency representation, could enable the generation of sounds not adhering strictly to twelve-tone pitch. This however, will require in both cases the creation and training of generative models on those types of music and sounds in order to properly assess the viability of these ideas.

Part IV

APPENDIX

A INFRASTRUCTURE AND ENGINEERING DETAILS

RATIONALE As part of this thesis, I deployed three interactive tools for both musicians and the general audience:

- NONOTO, for sheet music generation by inpainting,
- NOTONO, for visual transformation of musical instruments sounds via spectrogram inpainting,
- PIANOTO, for the generation and editing of expressive piano performances via inpainting.

The three interfaces share the same, open-source code-base, and are packaged together under the name `music-inpainting.ts`. The code is available on Sony CSL Paris' GitHub[1].

All three interfaces require connection to a compatible model for inpainting operations. They all essentially share the same API based on a single interactive `/inpaint` command, which takes as input a `media` (sheet, MIDI performance, sound...), which defines the base material for inpainting, and a `mask` (describing the portion of media to inpaint) and returns an updated version of the input media, with the selected, masked zone inpainted:

```
inpaint(media: T, mask: Timestamps) → inpainted_media: T
```

Other primitives can be derived from this elementary command, such as `/generate`, which generates data from scratch using an empty context:

```
generate(duration: Timestamp) → media: T
```

defined as:

```
inpaint(emptyContext, duration)
```

We now detail the relevant technical details for building and deploying the models and interfaces making up the prototypes presented in this manuscript, along with subjective, retrospective opinions on specific technologies. The presented technologies rely for a significant part on the AWS ecosystem of cloud solutions, but these should by no

1 github.com/SonyCSLParis/music-inpainting-ts

means be taken as hard constraints. The use of AWS tools emerged from practical reasons as part of the PhD, due to people in my team at Sony CSL having experience with these technologies. The following paragraphs should therefore not be interpreted as an endorsement of AWS, but rather as highlighting one of many approaches to deploying such systems. In particular, I try to highlight the concepts underlying the decisions rather, independently from the actual AWS services. Note furthermore that these solutions were developed in a self-taught fashion through research on technical blogs and Internet tutorials and are therefore by no means to be interpreted as a reference. Finally, the landscape for both front-end tooling and back-end/MLOps stacks is rapidly changing and some of the solutions presented here are likely to be obsolete soon, if not already.

We start by presenting the infrastructure for the front-end – in short, *TypeScript*-based *static* interfaces, then move on to present the relevant details for a reasonably automated and robust *PyTorch+Docker*-based deployment of the back-end inference models.

A.1 FRONTEND

In this section, we detail the languages, libraries and services used for building and deploying the front-end part of the presented prototypes. A core aspect is that our prototypes can be deployed as simple static webpages: the content of the interfaces, aside from the inpainted content, is completely known and determined in advance when bundling the JavaScript sources, and there is no need to retrieve data from e.g. an external database. This makes for cost-effective, easy to maintain deployments. Regarding the technologies, we rely on a simple, yet modern enough stack.

A.1.1 Technologies

The interfaces are all built with "pure" TypeScript. TypeScript is a statically typed superset of JavaScript, which simply adds type annotations, type inference and type-checking to the core JavaScript language and transparently compiles to standard JavaScript. The typing system provides many benefits, not just in terms of code robustness, but also by enabling powerful autocompletion and refactoring capabilities in compatible IDEs.

ON USING "VANILLA" TYPESCRIPT The interfaces were developed from the start without using any specific frontend framework (that is, no React, no Vue.js, no Svelte...). This was a personal choice out of pedagogical reasons: with no prior experience in web development or even interface development, I wanted to teach myself the core JavaScript/TypeScript language. I believe the pedagogical aspect has payed indeed, yet, due to this framework-free design, I also believe there is now room for large refactoring-related improvements in the current state of the codebase. Indeed, most of the reactive behaviors, that is, bindings of interactive aspects with their audio/visual rendering in the browser's view, are manually patched. This results in code that is arguably verbose if not convoluted. The codebase furthermore lacks automated testing suites. Still, all three interfaces have been battle-tested by the musicians with whom we collaborated at Sony CSL during the project and should prove rather stable at this point.

AUDIO LIBRARIES The interactive audio engine for all three interfaces is based on the `Tone.js` [127, 75] library, a powerful micro-framework for building DAW-like engines in the browser via the Web Audio API. Support for MIDI In and Out is implemented via the `webmidi` package [121, 25], which offers convenient abstractions over the Web MIDI API. Sheet display in NONOTO is handled by Open Sheet Music Display (OSMD) [124], an in-the-browser SVG sheet renderer based on the MusicXML sheet exchange format. Support for Ableton Link within the Electron packaging of NONOTO is provided via the Node.js bindings made available by Ishii [125].

A.1.2 Front-end deployment

The apps are fully *static*, except for the connection to the ML backend, therefore the frontend deployment is pretty straightforward: one should only serve the relevant, built HTML, CSS and JS files to users, without concerns such as user logins and authentication, database management...

Such static deployments therefore simply require a very basic HTTP server and can be achieved with cost-effective solutions like *AWS S3* (Simple Storage Service) or even for free via GitHub's *GitHub Pages* service. I have actually been using both, with AWS being used for the "private" links to specific interfaces, connected to the backend,

distributed to artists. The GitHub-Pages front-end deployment[2] does not contain links to the AWS hosted backend and instead prompts the users for a link to a compatible inference backend, for usage, e.g., along with a locally running Docker image of the relevant inference model.

AWS S3 *AWS S3* is the AWS service for static file hosting and distribution. In our case, AWS S3 offers an option to mark a storage unit (a 'bucket') as containing a static website, which then enables e.g. assigning a custom domain name to it. Custom domain names in turn are booked and managed with *Amazon Route 53*.

GITHUB PAGES GitHub Pages-based deployments can be done via GitHub CI and allow for automatic build-and-deploy on each version-tagging commit. In particular, since 2022, GitHub has added the possibility of uploading statically built scripts and assets via CI scripts independently of the codebase, instead of having to store the static website inside a specific branch as was previously done for deploying static websites. This avoids storing compiled files in the repository and represents a significant improvement. This solution is free for public, open-source repositories and can be conveniently automated via GitHub Actions (GitHub's solution for continuous integration), with relevant Action scripts available online.

A.2 ML BACKEND

We now detail the different technologies and services used for deploying the ML inference Back-End.

Thanks to the simple transformative operation described in the introduction of this chapter and around which the various models are designed performed by API, the APIs are designed to be completely *stateless*, that is, each request contains exactly all of the relevant information for its processing and does not require access to any external database or local state on the server (which could otherwise emerge in the form of cached passed information for e.g. autoregressive modelling). Focusing on statelessness makes for convenient micro-service-like deployments. Indeed, thanks to this, the running inference servers need not store any information from one request to

2 https://sonycslparis.github.io/music-inpainting-ts/

another, which amongst other things has the advantage of enabling easy-to-setup *load-balancing* of concurrent requests to multiple, identical and transparently replaceable instances. The advantages in terms of deployment transparency are significant, yet this somewhat redundant behavior could have a downside were the models to handle very large amounts of data such as video streams that would this have to be sent back and forth to the servers for every single inpainting operation. The prototypes presented here nevertheless do not significantly suffer from this limitation, indeed:

- NONOTO and PIANOTO operate purely on symbolic data, the required context information for inpainting operations is textual and therefore very lightweight,
- NOTONO processes audio and could therefore incur slowdown and costs from exchanging these audio files on each inpainting operation, yet by focusing on very short (4 seconds) sounds only, the overhead of retrieving the updated sound sample given an updated discrete latent codemap remains small.

CLOUD INSTANCES AWS-based deployments happen on so-called AWS EC2 (Elastic Cloud Compute) instances. Many different types of instances exist, with varying performances – parameters are CPU, RAM, access to GPUs and the type of GPUs... As of December 2022, the deployment for the various hosted models is done on `G4dn`-type instances, the most cost-effective NVidia GPU-based instance-type available on AWS. The `G4dn` instances are equipped with NVIDIA T4 GPUs, offering 16GB of GPU memory and are designed for running ML inference with lightweight models. In order to save costs, multiple models are served from each single GPU instance. Indeed, we can fit the DEEPBACH, NOTONO and PIA models on a single T4 GPU unit, which maximizes usage of each instance – at the cost of the models needing to share GPU bandwidth.

A.2.1 Technologies

The models are all developed in PyTorch and open-source. They are additionally distributed as standalone Docker images for easy re-use by interested practitioners. This enables users, provided that they have Docker installed on their local machine, to start runnning a local inference server for any of the models used in the prototypes,

in a single, simple command-line command. The relevant links are provided in the `music-inpainting.ts` GitHub repository.

All models are designed, trained and deployed within the PyTorch [87] framework, along with usage of the `torchaudio` library for audio-processing in NOTONO. Models are exported to TorchScript whenever possible for convenient, standalone distribution. Note however that TorchScript is apparently being phased-out as part of the move towards PyTorch 2.0[3].

Finally, the inference servers are deployed as simple Flask APIs directly running the PyTorch models, where Flask is the bare-bone solution for turning Python scripts into simple HTTP servers. This could be improved by using a dedicated *inference server* such as PyTorch's `torchserve`, NVidia's Triton or BentoML. This could for instance enable automatic batching of requests where relevant, and more generally could improve performances via relevant optimizations.

A.2.1.1 Docker

We use Docker for packaging and distributing the models. Docker enables easy transfer and reproducibility by packaging the model with all its relevant dependencies(both at the OS and Python levels) so that the resulting *containers*are standalone, self-contained units that can All models are packaged as Docker images and pushed onto a custom *AWS ECR* (Elastic Cloud Registry) registry, the AWS equivalent of the standard Docker Hub. This enables pulling and running models directly from the Docker Command-Line Interface (CLI) simply by referring their name and tag. For instance, the following command starts a local instance of the PIA inference server:

```
$ docker run -p 5000:5005 --rm --gpus 0 public.ecr.aws/csl-
    music-team/piano_inpainting_app:v3 serve
```

Relevant Docker commands for running the various models presented in this manuscript are provided on the project's GitHub repository at `https://github.com/SonyCSLParis/music-inpainting-ts`.

A.2.1.2 Container orchestration

To fully benefit from the automation brought by Docker-based deployments, one should introduce some form of container orchestration, that is, automated lifecycle management of instances, providing features

3 https://pytorch.org/get-started/pytorch-2.0/#technology-overview

such as automated detection and removal of unhealthy (e.g. unresponsive) instances and automated cluster scaling in response to variations in the incoming traffic. We chose to manage the deployments via *AWS Elastic Container Service*, which is the semi-managed orchestration service of AWS and was enough for the limited customization needed for our initial deploys.

HEALTH CHECKS The different Docker images all implement a simple health-check mechanism by periodically pinging the Flask server that they are supposed to serve to check if it is indeed alive. This enables *automatic restarts* of running containers whenever the Flask server stops being responsive[4]. These health-checks could be improved and made to be more semantic by replacing the simple ping mechanism with triggering actual inpainting requests (with dummy data) to actually check that the model can serve real requests and produces the expected outputs.

LAUNCH TEMPLATE AND THE GPU SHARING HACK To use ECS, one should define an *EC2 Auto Scaling Group* along with a *launch template* for EC2 instances (i.e. what type of instance, what firewall parameters...). Appropriate instances can then be launched automatically by ECS whenever needed. Note that the GPU sharing feature mentioned in the introduction to this section (and over which we rely to reduce deployment costs) is currently not officially supported on AWS ECS, yet there is a hack for enabling it at the instance level[5] on the GitHub repository for the AWS ML containers. This then permits running multiple models on a single, one-GPU instance, with all models sharing the GPU's memory.

Other existing solutions are *AWS Fargate* and *AWS EKS*. Fargate is AWS' fully-managed, turnkey solution to container orchestration, providing only very limited customizability, in particular would (I believe) not have made the GPU sharing hack mentioned above possible. On the other side of the spectrum, the fully customizable approach comes in the form of *AWS EKS* (Elastic Kubernetes Service), based on the Kubernetes orchestration system, appropriate for more advanced use-cases with finer control over, e.g., the auto-scaling procedure.

4 And is very convenient for peace of mind!

5 https://github.com/aws/containers-roadmap/issues/327#issuecomment-819792939

ENCAPSULATION: VPC All instances operate in an *AWS Virtual Private Cluster* (VPC), which accepts neither incoming nor outgoing connections to and from the Internet. This enables fine-grained control of the running instances and in particular helped in avoiding to receive malicious requests from hacker bots – which started arriving the very minute the instances were spawned when initial deploying them in a public-facing way! Indeed, the range of IP address reserved by AWS EC2 instances is fixed and hackers cycle over those to try and gain access to running instances. Some of those requests would have the Flask servers crash due to intentionally malformed input, therefore the encapsulation within a VPC proved valuable.

The API is therefore defined as an *AWS API Gateway* API which communicates with the backend ECS instances via an *AWS PrivateLink*, a *VPC link* which enables secure connection from the public-facing API Gateway to the secure VPC. connections happening via the API Gateway frontend.

A.2.2 On ML serving

AWS provides several technologies that claim to allow more cost-effective serving of CUDA-enabled models, yet my experience with those was not very convincing, with many of these new, cutting-edge technologies being more targeted at deploying very standard ML architectures and therefore failing to adapt to custom models. Due to these issues, I ended up giving up on trying to make use of these special technologies. Altogether, advance deep learning model compilers tend to be very bleeding-edge technologies, not all perfectly mature yet. This is for instance confirmed by the notes accompanying the announcement of PyTorch version 2.0[6], wherein the authors acknowledge the fact that TorchScript, which was pushed by the PyTorch team at Facebook for some years as the go-to solution for model deployment, was an ultimately imperfect and unsatisfactory approach, in particular in that it can require extensive model code modifications to enable compilation of custom models.

DISCLAIMER Note that this feedback is subjective and limited to my experience and that my inability to make these systems operate properly could also be attributed to me rather than the technologies themselves.

6 https://pytorch.org/get-started/pytorch-2.0/#technology-overview

AWS INFERENTIA AWS Inferentia chipsets are custom Amazon GPUs that are significantly cheaper than NVIDIA GPUs. The downside is that these chips require inference models to be compiled using a custom AWS compiler called AWS Neuron. I could never manage to make one of my models compile and the compilation times were dramatically long, along with never providing more detailed error feedback than: "there was a critical error, compilation aborted".

AWS ELASTIC INFERENCE AWS Elastic Inference is a solution for manually attaching partial GPUs to instances, with gradual memory: one should theoretically be able to attach 2Gb, 4Gb or 8Gb of GPU memory to an instance for accelerated ML inference. This, in theory, allows for cost-effective usage of GPU resources, by only provisioning the instances with exactly the right amount of GPU memory required by the models running on them. But Elastic Inference requires a custom AWS PyTorch runtime and the version distributed by Amazoe was always based on a PyTorch version significantly lagging behind the latest PyTorch stable release, which made it cumbersome to use when trying to deploy models relying on recent syntactic elements of the PyTorch language.

BIBLIOGRAPHY

[1] Agostinelli, A. *MusicLM: Generating Music From Text*. Jan. 26, 2023. DOI: 10.48550/arXiv.2301.11325. arXiv: 2301.11325 [cs, eess]. (Visited on 02/26/2023). preprint.

[2] Alisch, S. "'I Opened the Door to Develop Kuduro at JUPSON": Music Studios as Spaces of Collective Creativity in the Context of Electronic Dance Music in Angola." In: *Contemporary Music Review* 39.6 (Nov. 1, 2020), pp. 663–683. ISSN: 0749-4467. DOI: 10.1080/07494467.2020.1863004. (Visited on 01/04/2023).

[3] Aouameur, C., Esling, P., Hadjeres, G., "Neural Drum Machine : An Interactive System for Real-Time Synthesis of Drum Sounds." In: *Proceedings of the Tenth International Conference on Computational Creativity, ICCC 2019, Charlotte, North Carolina, USA, June 17-21, 2019*. Ed. by Kazjon Grace, Michael Cook, Dan Ventura, and Mary Lou Maher. Association for Computational Creativity (ACC), 2019, pp. 92–99. URL: http://computationalcreativity.net/iccc2019/papers/iccc19-paper-30.pdf.

[4] Assayag, G., Bloch, G., Chemillier, M., Cont, A., Dubnov, S., "OMAX Brothers: A Dynamic Topology of Agents for Improvisation Learning." In: *ACM Multimedia Workshop on Audio and Music Computing for Multimedia*. Santa Barbara, United States: Santa Barbara, 2006. URL: https://hal.inria.fr/hal-00839075 (visited on 11/21/2022).

[5] **Bazin, T.**, Hadjeres, G., "NONOTO: A Model-agnostic Web Interface for Interactive Music Composition by Inpainting." In: *Proceedings of the Tenth International Conference on Computational Creativity, ICCC 2019, Charlotte, North Carolina, USA, June 17-21, 2019*. Ed. by Kazjon Grace, Michael Cook, Dan Ventura, and Mary Lou Maher. Association for Computational Creativity (ACC), 2019, pp. 89–91. URL: http://computationalcreativity.net/iccc2019/papers/iccc19-paper-27.pdf (visited on 12/03/2021).

[6] **Bazin, T.**, Hadjeres, G., Esling, P., Malt, M., "Spectrogram Inpainting for Interactive Generation of Instrument Sounds."

In: *Proceedings of the 2020 Joint Conference on AI Music Creativity*. Stockholm, Norway: KTH Royal Institute of Technology, July 2020. DOI: 10.30746/978-91-519-5560-5. (Visited on 12/03/2021).

[7] **Bazin, T.**, Hadjeres, G., Malt, M., "AI-driven, Mobile-First Web UI for Controllable Expressive Piano Performance Composition." In: *Extended Abstracts for the Late-Breaking Demo Session of the 23rd Int. Society for Music Information Retrieval*. ISMIR Late-Breaking Demo Session. Bengaluru, India, 2022. URL: https://ismir2022program.ismir.net/lbd_395.html.

[8] Benaroya, L., Obin, N., Roebel, A., *Beyond Voice Identity Conversion: Manipulating Voice Attributes by Adversarial Learning of Structured Disentangled Representations*. July 27, 2021. DOI: 10.48550/arXiv.2107.12346. arXiv: 2107.12346 [cs, eess]. (Visited on 10/02/2022). preprint.

[9] Bitton, A., Esling, P., Harada, T., "Vector-Quantized Timbre Representation." In: *International Computer Music Conference*. Santiago, Chile, July 2021. URL: https://hal.archives-ouvertes.fr/hal-03208036 (visited on 12/28/2022).

[10] Bogaards, N. "Analysis-Assisted Sound Processing with Audiosculpt." In: 8th International Conference on Digital Audio Effects (DAFX-05). 2005, p. 269. URL: https://hal.science/hal-01161331 (visited on 03/04/2023).

[11] Bogaards, N., Röbel, A., Rodet, X., "Sound Analysis and Processing with AudioSculpt 2." In: *Proceedings of the 2004 International Computer Music Conference, ICMC 2004, Miami, Florida, USA, November 1-6, 2004*. Michigan Publishing, 2004. URL: https://hdl.handle.net/2027/spo.bbp2372.2004.131 (visited on 03/04/2023).

[12] Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Teboul, O., Grangier, D., Tagliasacchi, M., Zeghidour, N., *AudioLM: A Language Modeling Approach to Audio Generation*. Sept. 7, 2022. DOI: 10.48550/arXiv.2209.03143. arXiv: 2209.03143 [cs, eess]. (Visited on 02/26/2023). preprint.

[13] Bougueng Tchemeube, R., Ens, J. J., Pasquier, P., "Calliope: A Co-creative Interface for Multi-Track Music Generation." In: *Creativity and Cognition*. C&C '22. New York, NY, USA: Association for Computing Machinery, June 20, 2022, pp. 608–

611. ISBN: 978-1-4503-9327-0. DOI: 10.1145/3527927.3535200. (Visited on 10/01/2022).

[14] Boulanger-Lewandowski, N., Bengio, Y., Vincent, P., "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription." In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. ICML'12. Madison, WI, USA: Omnipress, June 26, 2012, pp. 1881–1888. ISBN: 978-1-4503-1285-1.

[15] Buzo, A., Jr. A. H. G., Gray, R. M., Markel, J. D., "A Two-Step Speech Compression System with Vector Quantizing." In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '79, Washington, D. C., USA, April 2-4, 1979*. IEEE, 1979, pp. 52–55. DOI: 10.1109/ICASSP.1979.1170755.

[16] Caillon, A., Esling, P., *RAVE: A Variational Autoencoder for Fast and High-Quality Neural Audio Synthesis*. Dec. 15, 2021. DOI: 10.48550/arXiv.2111.05011. arXiv: 2111.05011 [cs, eess]. (Visited on 10/01/2022). preprint.

[17] Cano, E., Beveridge, S., "Microtiming Analysis in Traditional Shetland Fiddle Music." In: *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*. Ed. by Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk. 2019, pp. 511–516. URL: http://archives.ismir.net/ismir2019/paper/000061.pdf (visited on 02/26/2023).

[18] Carter, S., Mankoff, J., Klemmer, S. R., Matthews, T., "Exiting the Cleanroom: On Ecological Validity and Ubiquitous Computing." In: *Human–Computer Interaction* 23.1 (Feb. 29, 2008), pp. 47–99. ISSN: 0737-0024. DOI: 10.1080/07370020701851086. (Visited on 03/05/2023).

[19] Cherry, E., Latulipe, C., "Quantifying the Creativity Support of Digital Tools through the Creativity Support Index." In: *ACM Trans. Comput. Hum. Interact.* 21.4 (2014), 21:1–21:25. DOI: 10.1145/2617588.

[20] Chowning, J. M. "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation." In: *Journal of the Audio Engineering Society* 21.7 (Sept. 1, 1973), pp. 526–534. URL: https://www.aes.org/e-lib/online/browse.cfm?elib=1954 (visited on 03/13/2023).

[21] *Coconet: The ML Model behind Today's Bach Doodle*. Magenta. URL: https://magenta.tensorflow.org/coconet (visited on 12/28/2022).

[22] Combes, A. B. *Comment l'université broie les jeunes chercheurs: Précarité, harcèlement, loi du silence*. Paris: AUTREMENT, Jan. 5, 2022. 336 pp. ISBN: 978-2-08-027047-4.

[23] Cook, P. R. "Principles for Designing Computer Music Controllers." In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. NIME 2001. Seattle, WA: Zenodo, June 1, 2001, pp. 3–6. DOI: 10.5281/zenodo.1176358. (Visited on 10/25/2022).

[24] Cooper, A., Reimann, R., Cronin, D., *About Face 3: The Essentials of Interaction Design*. USA: John Wiley & Sons, Inc., 2007. 648 pp. ISBN: 978-0-470-08411-3.

[25] Côté, J. P. "User-Friendly MIDI in the Web Browser." In: *International Conference on New Interfaces for Musical Expression*. NIME 2022. June 16, 2022. DOI: 10.21428/92fbeb44.388e4764. (Visited on 09/15/2022).

[26] Davies, M., Fuentes, M., Fonseca, J., Aly, L., Jerónimo, M., Baraldi, F. B., "Moving in Time: Computational Analysis of Microtiming in Maracatu de Baque Solto." In: *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*. Ed. by Julie Cumming, Jin Ha Lee, Brian McFee, Markus Schedl, Johanna Devaney, Cory McKay, Eva Zangerle, and Timothy de Reuse. 2020, pp. 795–802. URL: http://archives.ismir.net/ismir2020/paper/000113.pdf (visited on 02/26/2023).

[27] Deruty, E., Grachten, M., Lattner, S., Nistal, J., Aouameur, C., "On the Development and Practice of AI Technology for Contemporary Popular Music Production." In: *Transactions of the International Society for Music Information Retrieval* 5.1 (2022), p. 35. DOI: 10.5334/tismir.100.

[28] Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., Sutskever, I., *Jukebox: A Generative Model for Music*. Apr. 30, 2020. DOI: 10.48550/arXiv.2005.00341. arXiv: 2005.00341 [cs, eess, stat]. (Visited on 10/02/2022). preprint.

[29] Dieleman, S., Oord, A., Simonyan, K., "The Challenge of Realistic Music Generation: Modelling Raw Audio at Scale." In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett. 2018, pp. 8000–8010. URL: https://proceedings.neurips.cc/paper/2018/hash/3e441eec3456b703a4fe741005f3981f-Abstract.html.

[30] Donahue, C., Simon, I., Dieleman, S., *Piano Genie: An Intelligent Musical Interface*. Magenta. 2018. URL: https://magenta.tensorflow.org/pianogenie (visited on 12/28/2022).

[31] Donahue, C., Simon, I., Dieleman, S., "Piano Genie." In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. Mar. 17, 2019, pp. 160–164. DOI: 10.1145/3301275.3302288. arXiv: 1810.05246 [cs, eess, stat]. (Visited on 09/18/2022).

[32] Donahue, C. *SingSong: Generating Musical Accompaniments from Singing*. Jan. 29, 2023. DOI: 10.48550/arXiv.2301.12662. arXiv: 2301.12662 [cs, eess]. (Visited on 01/31/2023). preprint.

[33] Duignan, M., Noble, J., Barr, P., Biddle, R., "Metaphors for Electronic Music Production in Reason and Live." In: *Computer Human Interaction*. Ed. by Masood Masoodian, Steve Jones, and Bill Rogers. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, pp. 111–120. ISBN: 978-3-540-27795-8. DOI: 10.1007/978-3-540-27795-8_12.

[34] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., Norouzi, M., *Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders*. Apr. 5, 2017. DOI: 10.48550/arXiv.1704.01279. arXiv: 1704.01279 [cs]. (Visited on 09/18/2022). preprint.

[35] Engel, J. H., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C., Roberts, A., "GANSynth: Adversarial Neural Audio Synthesis." In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=H1xQVn09FX.

[36] Engel, J. H., Hantrakul, L., Gu, C., Roberts, A., "DDSP: Differentiable Digital Signal Processing." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa,*

Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL: https://openreview.net/forum?id=B1x1ma4tDr.

[37] Esling, P., **Bazin, T.**, Bitton, A., Carsault, T., Devis, N., "Ultra-Light Deep MIR by Trimming Lottery Tickets." In: International Symposium on Music Information Retrieval (ISMIR). Oct. 15, 2020. URL: https://hal.archives-ouvertes.fr/hal-03208026 (visited on 12/06/2021).

[38] Esling, P., Masuda, N., Bardet, A., Despres, R., Chemla-Romeu-Santos, A., "Flow Synthesizer: Universal Audio Synthesizer Control with Normalizing Flows." In: *Applied Sciences* 10.1 (Jan. 2020), p. 302. DOI: 10.3390/app10010302. (Visited on 10/02/2022).

[39] Esser, P., Rombach, R., Ommer, B., "Taming Transformers for High-Resolution Image Synthesis." In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, June 2021, pp. 12868–12878. ISBN: 978-1-66544-509-2. DOI: 10.1109/CVPR46437.2021.01268. (Visited on 05/19/2022).

[40] Faber, T. *Decolonizing Electronic Music Starts With Its Software*. Pitchfork. Feb. 25, 2021. URL: https://pitchfork.com/thepitch/decolonizing-electronic-music-starts-with-its-software/ (visited on 02/26/2023).

[41] Forsgren, S., Martiros, H., *Riffusion: Stable Diffusion for Real-Time Music Generation*. Riffusion. URL: http://www.riffusion.com (visited on 12/20/2022).

[42] Freed, A., Schmeder, A., "Features and Future of Open Sound Control Version 1.1 for NIME." In: *9th International Conference on New Interfaces for Musical Expression, NIME 2009, Pittsburgh, PA, USA, June 4-6, 2009*. nime.org, 2009, pp. 116–120. URL: http://www.nime.org/proceedings/2009/nime2009%5C_116.pdf (visited on 02/25/2023).

[43] Fuentes, M., Maia, L., Rocamora, M., Biscainho, L. W. P., Crayencour, H. C., Essid, S., Bello, J. P., "Tracking Beats and Microtiming in Afro-Latin American Music Using Conditional Random Fields and Deep Learning." In: *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*. Ed. by Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk. 2019,

pp. 251–258. URL: http://archives.ismir.net/ismir2019/paper/000029.pdf (visited on 02/26/2023).

[44] Gareus, R. "The Ardour DAW – Latency Compensation and Anywhere-to-Anywhere Signal Routing Systems." These de doctorat. Paris 8, Dec. 8, 2017. URL: https://www.theses.fr/2017PA080116 (visited on 02/28/2023).

[45] Goltz, F. "Ableton Link – A Technology to Synchronize Music Software." In: Linux Audio Conference 2018. Berlin, June 2018, p. 4. URL: https://lac.linuxaudio.org/2018/pdf/42-paper.pdf.

[46] Goodfellow, I., Bengio, Y., Courville, A., *Deep Learning*. MIT Press, 2016.

[47] Hadjeres, G., Pachet, F., Nielsen, F., "DeepBach: A Steerable Model for Bach Chorales Generation." In: *Proc. of the 34th International Conference on Machine Learning (ICML)*. 2017, pp. 1362–1371.

[48] Hadjeres, G., Crestel, L., *Vector Quantized Contrastive Predictive Coding for Template-based Music Generation*. Apr. 21, 2020. DOI: 10.48550/arXiv.2004.10120. arXiv: 2004.10120 [cs, eess]. (Visited on 10/02/2022). preprint.

[49] Hadjeres, G., Crestel, L., *The Piano Inpainting Application*. July 13, 2021. DOI: 10.48550/arXiv.2107.05944. arXiv: 2107.05944 [cs, eess]. (Visited on 09/15/2022). preprint.

[50] Hadjeres, G., Nielsen, F., "Anticipation-RNN: Enforcing Unary Constraints in Sequence Generation, with Application to Interactive Music Generation." In: *Neural Computing & Applications* 32.4 (2020), pp. 995–1005. DOI: 10.1007/s00521-018-3868-4.

[51] Hallström, E., Mossmyr, S., Sturm, B., Vegeborn, V., Wedin, J., "From Jigs and Reels to Schottisar Och Polskor : Generating Scandinavian-like Folk Music with Deep Recurrent Networks." In: The 16th Sound & Music Computing Conference, Malaga, Spain, 28-31 May 2019. 2019. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-248982 (visited on 03/17/2023).

[52] Hantrakul, H. *Creating Sounds Of India: An on Device, AI Powered, Musical Experience Built with TensorFlow*. TensorFlow Blog. 2020. URL: https://blog.tensorflow.org/2020/08/creating-sounds-of-india-with-tensorflow.html (visited on 12/28/2022).

[53] Hantrakul, H., Zada, N., Carney, M., Bowers, M., Li, C., Toh, E., Secor, J., Engel, J., *Tone Transfer*. Magenta. 2020. URL: https://magenta.tensorflow.org/tone-transfer (visited on 12/28/2022).

[54] *Harmonai-Org/Sample-Generator: Tools to Train a Generative Model on Arbitrary Audio Samples*. URL: https://github.com/Harmonai-org/sample-generator (visited on 12/26/2022).

[55] Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., Engel, J. H., Oore, S., Eck, D., "Onsets and Frames: Dual-Objective Piano Transcription." In: *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*. Ed. by Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos. 2018, pp. 50–57. URL: http://ismir2018.ircam.fr/doc/pdfs/19%5C_Paper.pdf (visited on 12/01/2022).

[56] Huang, C.-Z. A., Cooijmans, T., Roberts, A., Courville, A. C., Eck, D., "Counterpoint by Convolution." In: *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*. Ed. by Sally Jo Cunningham, Zhiyao Duan, Xiao Hu, and Douglas Turnbull. 2017, pp. 211–218. URL: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/187%5C_Paper.pdf (visited on 12/28/2022).

[57] Huang, C.-Z. A., Hawthorne, C., Roberts, A., Dinculescu, M., Wexler, J., Hong, L., Howcroft, J., "Approachable Music Composition with Machine Learning at Scale." In: *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, the Netherlands, November 4-8, 2019*. Ed. by Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk. 2019, pp. 793–800. URL: http://archives.ismir.net/ismir2019/paper/000097.pdf.

[58] Huang, C.-Z. A., Koops, H. V., Newton-Rex, E., Dinculescu, M., Cai, C., "AI Song Contest: Human-AI Co-Creation in Songwriting." In: *Proceedings of the 21st International Society for Music Information Retrieval Conference*. ISMIR. Montreal, Canada: ISMIR, Oct. 11, 2020, pp. 708–716. DOI: 10.5281/zenodo.4245530. (Visited on 01/29/2023).

[59] Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A. M., Hoffman, M. D., Dinculescu, M., Eck, D., "Music Transformer: Generating Music with Long-Term

Structure." In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=rJe4ShAcF7.

[60] Hunt, A., Wanderley, M. M., Paradis, M., "The Importance of Parameter Mapping in Electronic Instrument Design." In: *Journal of New Music Research* 32.4 (Dec. 1, 2003), pp. 429–440. ISSN: 0929-8215. DOI: 10.1076/jnmr.32.4.429.18853. (Visited on 01/12/2023).

[61] Isola, P., Zhu, J.-Y., Zhou, T., Efros, A. A., "Image-to-Image Translation with Conditional Adversarial Networks." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). July 2017, pp. 5967–5976. DOI: 10.1109/CVPR.2017.632.

[62] Jo, Y., Park, J., "SC-FEGAN: Face Editing Generative Adversarial Network With User's Sketch and Color." In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Oct. 2019, pp. 1745–1753. DOI: 10.1109/ICCV.2019.00183.

[63] Jump, B. *The Origin of Notation*. Chasing the Chords. Aug. 30, 2015. URL: https://brianjump.net/2015/08/29/the-origin-of-notation/ (visited on 01/12/2023).

[64] Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F., "Transformers Are RNNs: Fast Autoregressive Transformers with Linear Attention." In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5156–5165. URL: http://proceedings.mlr.press/v119/katharopoulos20a.html.

[65] Kemp, S. *The Global State of Digital in July 2022, Pt. 1*. We Are Social. July 26, 2022. URL: https://wearesocial.com/blog/2022/07/the-global-state-of-digital-in-july-2022-part-one/ (visited on 03/22/2023).

[66] Kilgour, K., Zuluaga, M., Roblek, D., Sharifi, M., "Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms." In: *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*. Ed. by Gernot Kubin and

Zdravko Kacic. ISCA, 2019, pp. 2350–2354. DOI: 10.21437/Interspeech.2019-2219.

[67] Kong, Q., Li, B., Chen, J., Wang, Y., "GiantMIDI-Piano: A Large-Scale MIDI Dataset for Classical Piano Music." In: *Transactions of the International Society for Music Information Retrieval* 5.1 (1 May 12, 2022), pp. 87–98. ISSN: 2514-3298. DOI: 10.5334/tismir.80. (Visited on 10/01/2022).

[68] Koszolko, M. "The Tactile Evolution: Electronic Music Production and Affordances of iOS Apps." In: *Proceedings of the 12th Art of Record Production Conference*. The Art of Record Production Conference. Vol. 12. Royal College of Music (KMH) & Art of Record Production, 2019, pp. 187–204. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kmh:diva-3130 (visited on 02/05/2023).

[69] Le Moine, C., Obin, N., Roebel, A., "Towards End-to-End Fo Voice Conversion Based on Dual-GAN with Convolutional Wavelet Kernels." In: *2021 29th European Signal Processing Conference (EUSIPCO)*. 2021 29th European Signal Processing Conference (EUSIPCO). Aug. 2021, pp. 36–40. DOI: 10.23919/EUSIPCO54536.2021.9616190.

[70] Leger, R. "Schnittstellen zur interaktiven Nutzung von Künstlicher Intelligenz in der Musikproduktion." Master's Degree. Universität Bayreuth, 2021.

[71] Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., Catanzaro, B., "Image Inpainting for Irregular Holes Using Partial Convolutions." 2018. arXiv: 1804.07723.

[72] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J., "On the Variance of the Adaptive Learning Rate and Beyond." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rkgz2aEKDr.

[73] Louie, R., Coenen, A., Huang, C. Z., Terry, M., Cai, C. J., "Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. New York, NY, USA: Association for Computing Machinery, Apr. 23, 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376739. (Visited on 10/31/2022).

[74] Lumumba-Kasongo, E. *(A)I, Rapper: Who Voices Hip-Hop's Future?* Public Books. Apr. 21, 2022. URL: https://www.publicbooks.org/ai-rap-synthesis-tools-black-hip-hop/ (visited on 02/26/2023).

[75] Mann, Y. "Interactive Music with Tone.Js." In: *First Web Audio Conference*. Web Audio Conference. Paris, France, Jan. 26, 2015, p. 5. URL: https://wac.ircam.fr/pdf/wac15_submission_40.pdf.

[76] Michel, E., Boubekeur, T., "DAG Amendment for Inverse Control of Parametric Shapes." In: *ACM Transactions on Graphics* 40.4 (July 19, 2021), 173:1–173:14. ISSN: 0730-0301. DOI: 10.1145/3450626.3459823. (Visited on 09/16/2022).

[77] Müller, R., Kornblith, S., Hinton, G. E., "When Does Label Smoothing Help?" In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. URL: https://papers.nips.cc/paper/2019/hash/f1748d6b0fd9d439f71450117eba2725-Abstract.html (visited on 10/02/2022).

[78] *Musical Notation - Evolution of Western Staff Notation | Britannica*. URL: https://www.britannica.com/art/musical-notation/Evolution-of-Western-staff-notation (visited on 01/12/2023).

[79] Nistal, J., Aouameur, C., Lattner, S., Richard, G., "VQCPC-GAN: Variable-Length Adversarial Audio Synthesis Using Vector-Quantized Contrastive Predictive Coding." In: *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). Oct. 2021, pp. 116–120. DOI: 10.1109/WASPAA52581.2021.9632757.

[80] Nistal Hurle, J., Lattner, S., Richard, G., "DrumGAN: Synthesis of Drum Sounds with Timbral Feature Conditioning Using Generative Adversarial Networks." In: *21 St International Society for Music Information Retrieval Conference (ISMIR)*. Toronto, Canada, Aug. 2020. URL: https://hal.telecom-paris.fr/hal-03233337 (visited on 12/30/2022).

[81] Nistal Hurlé, J. "Exploring Generative Adversarial Networks for Controllable Musical Audio Synthesis." PhD thesis. Institut Polytechnique de Paris, Mar. 9, 2022. DOI: 10/document. (Visited on 12/30/2022).

[82] Nous Sommes Whim Therapy, *Nous Sommes Whim Therapy | AI Song Contest 2021 Entry Page*. AI Song Contest. 2021. URL: https://www.aisongcontest.com/participants/noussommeswhimtherapy-2021 (visited on 03/06/2023).

[83] Olah, C. *Neural Networks, Manifolds, and Topology*. colah's blog. Apr. 6, 2014. URL: https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/ (visited on 12/29/2022).

[84] Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., *WaveNet: A Generative Model for Raw Audio*. Sept. 19, 2016. DOI: 10.48550/arXiv.1609.03499. arXiv: 1609.03499 [cs]. (Visited on 10/02/2022). preprint.

[85] OpenAI, *ChatGPT: Optimizing Language Models for Dialogue*. OpenAI. Nov. 30, 2022. URL: https://openai.com/blog/chatgpt/ (visited on 02/26/2023).

[86] Papadopoulos, A., Roy, P., Pachet, F., "Assisted Lead Sheet Composition Using FlowComposer." In: *Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Toulouse, France, September 5-9, 2016, Proceedings*. Ed. by Michel Rueher. Vol. 9892. Lecture Notes in Computer Science. Springer, 2016, pp. 769–785. DOI: 10.1007/978-3-319-44953-1_48.

[87] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., "Automatic Differentiation in PyTorch." In: *NIPS-W*. 2017.

[88] Qian, K., Zhang, Y., Chang, S., Cox, D., Hasegawa-Johnson, M., "Unsupervised Speech Decomposition via Triple Information Bottleneck." In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, July 13, 2020, pp. 7836–7846.

[89] Razavi, A., Oord, A., Vinyals, O., "Generating Diverse High-Fidelity Images with VQ-VAE-2." In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 14837–14847. URL: https://proceedings.neurips.cc/paper/2019/hash/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Abstract.html.

[90] Remy, C., MacDonald Vermeulen, L., Frich, J., Biskjaer, M. M., Dalsgaard, P., "Evaluating Creativity Support Tools in HCI Research." In: *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. DIS '20: Designing Interactive Systems Conference 2020. Eindhoven Netherlands: ACM, July 3, 2020, pp. 457–476. ISBN: 978-1-4503-6974-9. DOI: 10.1145/3357236.3395474. (Visited on 07/15/2022).

[91] Reynolds, S. *How Auto-Tune Revolutionized the Sound of Popular Music*. Pitchfork. 2018. URL: https://pitchfork.com/features/article/how-auto-tune-revolutionized-the-sound-of-popular-music/ (visited on 01/12/2023).

[92] Roberts, A., Engel, J., Mann, Y., Gillick, J., Kayacik, C., Nørly, S., Dinculescu, M., Radebaugh, C., Hawthorne, C., Eck, D., "Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live." In: *Proceedings of the International Workshop on Musical Metacreation (MUME)*. 2019. URL: http://musicalmetacreation.org/buddydrive/file/mume_2019_paper_2/ (visited on 09/18/2022).

[93] Roberts, A., Engel, J. H., Raffel, C., Hawthorne, C., Eck, D., "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music." In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4361–4370. URL: http://proceedings.mlr.press/v80/roberts18a.html.

[94] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., *High-Resolution Image Synthesis with Latent Diffusion Models*. Apr. 13, 2022. DOI: 10.48550/arXiv.2112.10752. arXiv: 2112.10752 [cs]. (Visited on 09/17/2022). preprint.

[95] Rouard, S., Hadjeres, G., "CRASH: Raw Audio Score-Based Generative Modeling for Controllable High-Resolution Drum Sound Synthesis." In: *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*. Ed. by Jin Ha Lee, Alexander Lerch, Zhiyao Duan, Juhan Nam, Preeti Rao, Peter van Kranenburg, and Ajay Srinivasamurthy. 2021, pp. 579–585. URL: https://archives.ismir.net/ismir2021/paper/000072.pdf.

[96] Sauer, J., Sonderegger, A., "The Influence of Product Aesthetics and User State in Usability Testing." In: *Behaviour & Information Technology* 30.6 (Nov. 1, 2011), pp. 787–796. ISSN: 0144-929X. DOI: 10.1080/0144929X.2010.503352. (Visited on 03/04/2023).

[97] Scurto, H. "Designing With Machine Learning for Interactive Music Dispositifs." These de doctorat. Sorbonne université, Dec. 2, 2019. URL: https://www.theses.fr/2019SORUS356 (visited on 11/21/2022).

[98] Serrà, J., Koduri, G. K., Miron, M., Serra, X., "Assessing the Tuning of Sung Indian Classical Music." In: *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*. Ed. by Anssi Klapuri and Colby Leider. University of Miami, 2011, pp. 157–162. URL: http://ismir2011.ismir.net/papers/OS2-2.pdf (visited on 02/26/2023).

[99] Sheridan, G. "Fruity Batidas: The Technologies and Aesthetics of Kuduro." In: *Dancecult* 6.1 (2014), pp. 83–96. ISSN: 19475403. DOI: 10.12801/1947-5403.2014.06.01.05. (Visited on 01/04/2023).

[100] *Shruti (Music)*. In: *Wikipedia*. Mar. 12, 2023. URL: https://en.wikipedia.org/wiki/Shruti_(music) (visited on 03/24/2023).

[101] Silpayamanant, J. *DAW, Music Production, and Colonialism, a Bibliography*. Mae Mai. Feb. 25, 2021. URL: https://silpayamanant.wordpress.com/bibliography/daw-colonialism/ (visited on 02/26/2023).

[102] Strachan, R. "Affordances, stations audionumériques et créativité musicale." In: *Réseaux* 172.2 (2012), pp. 120–143. ISSN: 0751-7971. DOI: 10.3917/res.172.0120. (Visited on 02/05/2023).

[103] Sturm, B., Santos, J. F., Korshunova, I., "Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units." In: *16th International Society for Music Information Retrieval Conference, Late-Breaking Demo Session*. 16th International Society for Music Information Retrieval Conference. 2015. URL: http://hdl.handle.net/1854/LU-7053070 (visited on 03/17/2023).

[104] Taruskin, R. *Oxford History of Western Music*. Oxford, New York: Oxford University Press, 2005.

[105] Tay, Y., Dehghani, M., Bahri, D., Metzler, D., "Efficient Transformers: A Survey." In: *ACM Computing Surveys* 55.6 (Dec. 7, 2022), 109:1–109:28. ISSN: 0360-0300. DOI: 10.1145/3530811. (Visited on 03/11/2023).

[106] Taylor, B., Bernstein, A., *Tune.Js: A Microtonal Web Audio Library*. Georgia Institute of Technology, Apr. 2016. ISBN: 978-0-692-61973-5. URL: https://smartech.gatech.edu/handle/1853/54580 (visited on 09/26/2022).

[107] Theis, L., Oord, A., Bethge, M., "A Note on the Evaluation of Generative Models." In: *International Conference on Learning Representations*. Apr. 2016. URL: http://arxiv.org/abs/1511.01844.

[108] Oord, A., Vinyals, O., Kavukcuoglu, K., "Neural Discrete Representation Learning." In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. 2017, pp. 6306–6315. URL: https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

[109] Oord, A. "Parallel WaveNet: Fast High-Fidelity Speech Synthesis." In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 3915–3923. URL: http://proceedings.mlr.press/v80/oord18a.html.

[110] Viraraghavan, V. S., Aravind, R., Murthy, H. A., "Precision of Sung Notes in Carnatic Music." In: *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*. Ed. by Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos. 2018, pp. 499–505. URL: http://ismir2018.ircam.fr/doc/pdfs/120%5C_Paper.pdf (visited on 02/26/2023).

[111] Wanderley, M. M., Mackay, W. E., "HCI, Music and Art: An Interview with Wendy Mackay." In: *New Directions in Music and Human-Computer Interaction*. Ed. by Simon Holland, Tom Mudd, Katie Wilkie-McKenna, Andrew P. McPherson, and Marcelo M.

Wanderley. Springer, 2019, pp. 115–120. DOI: 10.1007/978-3-319-92069-6_7. (Visited on 02/05/2023).

[112] Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R. J., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., Saurous, R. A., "Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis." In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 5167–5176. URL: http://proceedings.mlr.press/v80/wang18h.html.

[113] Weng, L. *What Are Diffusion Models?* July 11, 2021. URL: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/ (visited on 02/04/2023).

[114] Wessel, D., Wright, M., "Problems and Prospects for Intimate Musical Control of Computers." In: *Computer Music Journal* 26.3 (Sept. 1, 2002), pp. 11–22. ISSN: 0148-9267. DOI: 10.1162/014892602320582945.

[115] Wessel, D., Wright, M., "2001: Problems and Prospects for Intimate Musical Control of Computers." In: *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*. Ed. by Alexander Refsum Jensenius and Michael J. Lyons. Current Research in Systematic Musicology. Cham: Springer International Publishing, 2017, pp. 15–27. ISBN: 978-3-319-47214-0. DOI: 10.1007/978-3-319-47214-0_2.

[116] Wright, M., Freed, A., "Open SoundControl: A New Protocol for Communicating with Sound Synthesizers." In: *Proceedings of the 1997 International Computer Music Conference*. ICMC. Thessaloniki, Greece: Michigan Publishing, Sept. 25–30, 1997. URL: https://hdl.handle.net/2027/spo.bbp2372.1997.033 (visited on 02/25/2023).

[117] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T., "Free-Form Image Inpainting With Gated Convolution." In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Oct. 2019, pp. 4470–4479. DOI: 10.1109/ICCV.2019.00457.

SOFTWARE

[118] AUTOMATIC1111, *Stable Diffusion Web UI*. Sept. 18, 2022. URL: https://github.com/AUTOMATIC1111/stable-diffusion-webui (visited on 09/18/2022).

[119] **Bazin, T.**, Hadjeres, G., *SonyCSLParis/Music-Inpainting.Ts*. Version 1.4.6. Oct. 2022. DOI: 10.5281/zenodo.7263101. (Visited on 11/01/2022).

[120] Cífka, O. *Html-Midi-Player*. Version 1.5.0. July 9, 2022. URL: https://github.com/cifkao/html-midi-player/.

[121] Côté, J. P. *WebMidi.Js*. Version 3.0.21. 2015. URL: https://github.com/djipco/webmidi.

[122] *Electron/Electron*. Electron, Dec. 24, 2022. URL: https://github.com/electron/electron (visited on 12/24/2022).

[123] Engel, J. *DDSP-VST*. In collab. with Wilson Zhao, Nikhil Bhanu, Megan Jurek, and Yury Kartynnik. Magenta, Dec. 23, 2022. URL: https://github.com/magenta/ddsp-vst (visited on 12/28/2022).

[124] Haas, S., Schmid, S., Uiberacker, M., Giesinger, B., *OpenSheetMusicDisplay (OSMD)*. Open Sheet Music Display, Dec. 16, 2022. URL: https://github.com/opensheetmusicdisplay/opensheetmusicdisplay (visited on 12/19/2022).

[125] Ishii, T. *Node-Abletonlink*. Dec. 22, 2022. URL: https://github.com/2bbb/node-abletonlink (visited on 12/24/2022).

[126] Magenta, *@magenta/Music*. In collab. with Monica Dinculescu, Adam Roberts, Ian Simon, Curtis "Fjord" Hawthorne, and contributors. Version 1.23.1. Google, 2021. URL: https://github.com/magenta/magenta-js/tree/master/music.

[127] Mann, Y. *Tone.Js*. Tonejs, 2014. URL: https://github.com/Tonejs/Tone.js (visited on 09/22/2022).

[128] Mann, Y. *A.I. Duet*. 2017. URL: https://experiments.withgoogle.com/ai-duet (visited on 09/15/2022).

[129] Mann, Y. *Googlecreativelab/Aiexperiments-Ai-Duet*. Google Creative Lab, 2017. URL: https://github.com/googlecreativelab/aiexperiments-ai-duet (visited on 09/25/2022).

VIDEO RECORDINGS

[130] Sony CSL, director. *Uele Lamore's Collaboration with the "Music & Artificial Intelligence" Team of Sony CSL*. In collab. with Uèle Lamore. Apr. 1, 2021. URL: https://www.youtube.com/watch?v=H12OA2rjWeg (visited on 03/05/2023).

[131] Sony CSL, director. *Whim Therapy Collaborates with Sony CSL for the AI Song Contest 2021*. In collab. with Whim Therapy. June 14, 2021. URL: https://www.youtube.com/watch?v=pGtrHuYN7Kw (visited on 03/06/2023).

ATTRIBUTIONS

Click icon on Figure 4.1b by Nociconist from Noun Project published under a Creative Commons license.

Other external images have been borrowed to illustrate this thesis. Sources are mentioned in the accompanying captions.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/