Meaningful audio synthesis and musical interactions by representation learning of sound sample databases.

Adrien Bitton

Meaningful audio synthesis and musical interactions by representation learning of sound sample databases

Ph.D. thesis in computer science

Adrien BITTON
bitton@ircam.fr

Institut de Recherche et Coordination Acoustique Musique (IRCAM)
Sciences et technologies de la musique et du son (UMR9912)
1, Place Igor Stravinsky, F-75004 Paris, France
Ecole Doctorale Informatique, Télécommunications
et Electronique (ED130)
Sorbonne Université, Paris-VI

Director: Pr. Carlos AGON
Supervisor: Pr. Philippe ESLING

Defended on the 14th of June 2021 at IRCAM, Paris.

When we accept small wonders,
we qualify ourselves to imagine great wonders.
In **Jitterbug Perfume**, by **Tom Robbins**.

Jury committee

- ▷ **Pr. Philippe PASQUIER**, Simon Fraser University, Canada (reviewer).
- ▷ **Pr. Charalampos SAITIS**, Queen Mary University of London, United Kingdom (reviewer).
- ▷ **Pr. Jean-Pierre BRIOT**, Sorbonne Université, France.
- ▷ **Pr. Myriam DESAINTE-CATHERINE**, Université de Bordeaux, France.
- ▷ **Pr. Dorien HERREMANS**, Singapore University of Technology and Design, Singapore.
- ▷ **Pr. Bob L. T. STURM**, Royal Institute of Technology KTH, Sweden.

Invitees

- ▷ **Pr. Suguru GOTO**, Tokyo University of the Arts, Japan.
- ▷ **Pr. Tatsuya HARADA**, The University of Tokyo, Japan.

Abstract

Computer assisted music extensively relies on audio sample libraries and virtual instruments which provide users an ever increasing amount of contents to produce music with. However, principled methods for large-scale interactions are lacking so that browsing samples and presets with respect to a target sound idea is a tedious and arbitrary process. Indeed, library metadata can only describe coarse categories of sounds but do not meaningfully traduce the underlying acoustic contents and continuous variations in timbre which are key elements of music production and creativity. Timbre perception has been studied by carrying listening tests and organising human ratings into low dimensional spaces which reflect the perceived similarity of sounds, however these analysis spaces do not generalise to new and unrated examples, nor they allow to synthesise audio.

Digital signal processing models have been applied to analysis and synthesis, so that the extracted parameters can be manipulated and inverted back to audio. However, we observe that these methods require a high number of parameters and representation dimensions to allow accurate reconstructions. Visualisation and control are thus little intuitive, moreover these invertible dimensions mainly correspond to low-level signal properties and do not represent much semantic information. The recent advances in deep generative modelling show unprecedented successes at learning large-scale unsupervised representations which invert to data as diverse as images, texts and audio. These probabilistic models could be refined to specific generative tasks such as unpaired image translation and semantic manipulations of visual features, demonstrating the ability of learning transformations and representations that are perceptually meaningful.

The application of deep generative models to musical audio is at early stages and requires adapted model architectures and interactions. High quality auto-regressive waveform synthesis has been achieved for both speech and musical audio, however these models are computationally intensive, unsuited to moderate dataset sizes and offer little control over the generation for creative purposes. In this thesis, we target efficient analysis and synthesis with auto-encoders to learn low dimensional acoustic representations for timbre manipulations and intuitive interactions for music production. We adapt domain translation techniques to timbre transfer and propose alternatives to adversarial learning for many-to-many transfers. In this process, timbre is implicitly modelled by disentangling the representations of domain specific and domain invariant features. Then we develop models for explicit modelling of timbre variations and controllable audio sampling using conditioning for semantic attribute manipulations and hierarchical learning to represent both acoustic and temporal variations. We also apply discrete representation learning to decompose a target timbre into short-term acoustic features that are applied to audio conversions such as timbre transfer and voice-

driven synthesis. By analysing and mapping this discrete latent representation, we show that we can directly control synthesis by acoustic descriptors. Finally, we investigate the possibility of further reducing the complexity of trained models by weight trimming for real-time inference with constrained computational resources. Because the objectives used for training the models are often disjoint from the ultimate generative application, our discussion and evaluation emphasise both aspects of learning performance and usability as a creative tool for music production.

The organisation of this thesis is as follows. The first section introduces computational music processing, its different levels of expression and sets the problem domain of meaningful audio synthesis for music and with machine learning tools. The second section details the representation properties of audio and music information, it reviews classical models for audio synthesis and introduces the data-driven approach to music processing with machine learning. The third section introduces unsupervised learning and reviews the fundamentals of generative modelling with deep learning. The fourth section discusses related works in the field of neural audio synthesis with an emphasis in music applications. The fifth section presents the experiments carried during this thesis and discusses the results and evaluations, in the format of a "thesis by publications". The sixth section summarizes the results and publications along with related projects that were carried during the thesis and conclude with future works.

Résumé

La musique assistée par ordinateur fait beaucoup usage de librairies d'échantillons audios et d'instruments numériques qui offrent des possibilités de composition sans précédent. Cependant, l'abondance des matériaux sonores disponibles nécessite de nouvelles méthodes d'interaction en adéquation avec ceux-ci sans quoi le parcours des échantillons et configurations audios est inefficace et arbitraire. En effet, les métadonnées qui structurent traditionnellement ces librairies ne peuvent que traduire grossièrement les caractéristiques acoustiques des différentes catégories sonores. Notamment, les variations continues du timbre musical ne sont pas exprimées alors qu'elles jouent un rôle significatif dans la production et la créativité musicale. La perception du timbre a été étudiée par des testes d'écoute et l'analyse de ces résultats a permis la construction d'espaces de timbre dont les dimensions traduisent la similarité perceptive des différents sons. Cependant, ces espaces ne permettent pas d'analyser de nouveaux échantillons sonores et ils n'offrent aucun mécanisme inverse pour la génération audio.

Les modèles de traitement du signal numérique permettent l'analyse et la synthèse, de telle manière que les paramètres extraits du son peuvent être manipulés pour la production de nouveaux sons. Bien que ces techniques soient performantes, elles nécessitent souvent l'ajustement de nombreux paramètres afin d'obtenir des reconstructions précises et leur visualisation est ardue de part leurs représentations à haute dimensionnalité. Ainsi, le contrôle des techniques basées sur le traitement du signal manque d'intuitivité et les dimensions de ces espaces de synthèse sont principalement liées à des propriétés de bas niveau du signal qui ont une valeur sémantique limitée. Les progrès des modèles d'apprentissage génératif ont démontré des capacités sans précédent pour le traitement des données à grande échelle. Ces méthodes probabilistes permettent la construction d'espaces non supervisés pour la synthèse de données telles que les images, le texte ou le son et ont permis de nouvelles interactions telles que la conversion automatique d'images et la manipulation d'attributs perceptifs et stylistiques.

L'application des modèles d'apprentissage profond pour la génération audio a pris un essor au cours des dernières années et ce développement requiert des architectures adaptées ainsi que la conception d'interactions spécifiquement pensées pour la synthèse sonore. La synthèse directe de forme d'onde par des processus auto-régressifs a établi l'état de l'art pour la production de la voix et des sons musicaux. Bien qu'ils atteignent une haute qualité, ces modèles requièrent des puissances de calcul prohibitives et ne sont pas efficaces sur des bases de données de tailles limitées. De plus, les mécanismes auto-régressifs ont une modélisation locale performante mais leurs représentations et interactions sur les propriétés à long terme sont limitées. Au cours de cette thèse, nous développons des techniques d'analyse/synthèse

efficaces basées sur les modèles auto-encodeurs afin d'apprendre des représentations acoustiques inversibles de basse dimensionnalité pour la manipulation intuitive du timbre musical. En premier lieu, nous adaptons les techniques non supervisées de conversion d'images au transfert de propriétés de timbre. Nous proposons des objectifs alternatifs à l'entrainement par réseaux antagonistes génératifs qui permettent le transfert entre de multiples domaines, tels que des collections d'échantillons audios de différents instruments. Nous référons à cette approche comme une modélisation implicite du timbre qui est définit comme l'ensemble des propriétés qui ne sont pas partagées entre les différents domaines sonores. Ensuite, nous introduisons de nouveaux modèles pour l'apprentissage explicite de représentations du timbre musical et l'échantillonnage avec contrôle des propriétés acoustiques et sémantiques. Ces modèles s'appuient notamment sur le conditionnement du réseau génératif (décodeur) par des attributs musicaux cibles et l'apprentissage hiérarchique de représentations acoustiques locales et séquentielles à plus long terme. De plus, nous appliquons l'apprentissage de représentation discrète pour la décomposition acoustique du timbre qui permet de quantifier et convertir d'autres sources audios par reconstruction avec les propriétés de timbre apprisent dans le domaine cible. Ce faisant, nous proposons une méthode d'analyse de cette représentation discrète par descripteurs acoustiques qui permet le contrôle direct de la synthèse de variations acoustiques cibles. Enfin, nous avons conduit une étude sur la réduction des modèles d'apprentissage profond pour le traitement et la synthèse audio qui permet de réduire drastiquement la taille et le cout de calcul nécessaires à leur déploiement sur des systèmes grand-public et embarqués. Ainsi, notre discussion et évaluation ne se concentrent pas seulement sur la performance d'apprentissage mais aussi sur les qualités d'interaction et l'efficacité de ces modèles pour un usage avec des ressources de calcul contraintes.

L'organisation de cette thèse s'articule de la manière suivante. La première section introduit le traitement numérique de la musique, ses différents niveaux d'expression et pose la problématique de la synthèse audio avec les techniques d'apprentissage automatique. La seconde section détaille les propriétés des représentations de l'information musicale et sonore. Pour ce faire, nous récapitulons les méthodes classiques d'analyse et de synthèse ainsi que l'introduction des approches d'apprentissage. La troisième section détaille les fondements de l'apprentissage non supervisé et les principaux modèles génératifs de la littérature. La quatrième section détaille les tâches et modèles de référence appliqués à la synthèse audio musicale. La cinquième section fait un compte rendu des expériences effectuées au cours de la thèse, les contributions et résultats sont alors présentés dans le format d'une "thèse par articles". Enfin la sixième et dernière section conclut le manuscrit avec un résumé des travaux de recherche effectués, une discussion des projets conduits en parallèle de la thèse et les directions futures de recherche.

Acknowledgments

I would like to thank my thesis director Pr. Carlos Agon for his steady support throughout these years, with the calm and pleasant mood he puts into work. Then I would like to express my gratitude to Pr. Philippe Esling for his supervision, with passion and a unique twist he has put on many levels which go beyond that of the work. In my master's degree year at IRCAM, his research, teaching and humanity has inspired me into tackling this thesis topic and since then has not ceased to spark ideas within my work as well as that of the wonderful ACIDS team he has built within IRCAM. Many thanks to my workmates who all contributed to the good spirit in which we have been carrying our research, starting from the "oldest" members, Léopold, Axel, Tristan, Mathieu, Constance, Théis, Antoine and Ninon.

As I have been going around, I would like to thank the many foreign supports I have received. I would like to thank Pr. Tatsuya Harada for his invitation and outstanding support into the seven-month research I spent at the Machine Intelligence Laboratory in Tokyo. This was made possible by the short-term fellowship I received from the Japan Society for the Promotion of Science as well as the support I received at the University of Tokyo thanks to its many valued members, amongst whom, Antonio, Kurose, Yamane, Georges, Lisa and Hayato. I would also like to thank Pr. Suguru Goto from the Tokyo University of the Arts for his enthusiasm in organising with me a workshop on artificial intelligence and music. Moreover, I would like to thank Pr. Stefan Weinzierl at the Technical University of Berlin and Pr. Christoph Pörschmann at the University of Applied Sciences of Cologne for kindly receiving me during the work-time I spent in Germany.

By the end of my thesis, I was offered a position at Pixtunes GmbH in Berlin and I would like to thank my current employers Albrecht Panknin and Justus Klocke for their trust and the flexibility they allowed me to carry on the thesis writing while working. Many thanks as well to my kind and talented workmates in the artificial intelligence team, Thomas, Laura, Elliot and Marina.

None of this would have ever happened without the unconditional love and support of my family who has never failed at trusting in me, way beyond the limits of my own self-confidence. Neither it would have happened without the deep friendship bounds that created through life, which are invaluable and that I hold for the life time too. Amongst them, Valentin and Mathieu since our early times in Berlin, Pierre with whom I hope to run this year half-marathon in Berlin, Damian with whom I

collaborated on a few side projects during the thesis and Dara for your wise presence back in Edinburgh and all the way up to now.

Gratefulness may not fit into words, nor it would into names. To the countless persons who contribute(d) to my human development, friends, lovers, strangers met travelling, hitch-hiking, couch-surfing and in the wilderness of our marvellous world. I am thankful to all of you.

Lastly, I am extremely thankful to the jury members and reviewers who accepted with enthusiasm and kindness to take part in my defence, offering me valuable time and expertise needed to complete this thesis.

Contents

List of Figures

1 Introduction

1.1 Modalities of Music

The processing and storage of music span three main media which are *lyrics*, *score* and *sound*. The lyrics are structured texts which specify the language content of a song that is to be spoken or sung by human voice. It can be segmented in phrases and assigned to different voices or ensembles (e.g. groups in a choir). The score is a symbolic notation for music composition which specifies the notes to be played by the different instruments (e.g. melodies, chords). In order to be written and executed, it requires a specific musical understanding of harmony (tonality, keys, scales), rhythm (tempo, measures, subdivisions and signatures) as well as other parameters pertaining to the composition technique. For instance, the singing melodic content of a music can be defined with the score regardless of whether it comprises spoken words or not. This also applies to indicating specific playing modes and interpretation styles corresponding to a given instrument. The sound is the acoustic realisation of all the composition elements of music rendered into an audio signal by the performance process (instruments, singing voices). Although lyrics and scores can be read, sound is the universal sensory medium to experience music. This signal may take several forms, whether is is propagated in the listener auditory system, recorded or processed in the digital domain. Although some forms may be inherently lossy (e.g. recording medium, digital sampling), the audio signal is the richest modality of music that implicitly stores all of its components (Figure 1). Because audio is the medium that can ultimately be played-back and listened, it is the most widespread mean of music diffusion. It is a concrete realisation of music that does not require any particular skills to be sensed and enjoyed. At first we may divide music into the two main modalities of symbolic information and acoustic information. This reduction helps understanding some key properties of music, although it surely does not account for all kind of music practices. For instance, improvisation, live performance, noise music and many other facets of the ever changing field of music.

The symbolic information is an abstract medium of music processing. It is often composed of several layers of discrete elements (e.g. note onsets, pitches, words) that are structured according to conventions in music notation and sparsely distributed in time and classes. The score compresses the musical idea into series of explicit elements that can be visualised all at once. Due to this symbolic reduction, the score highlights

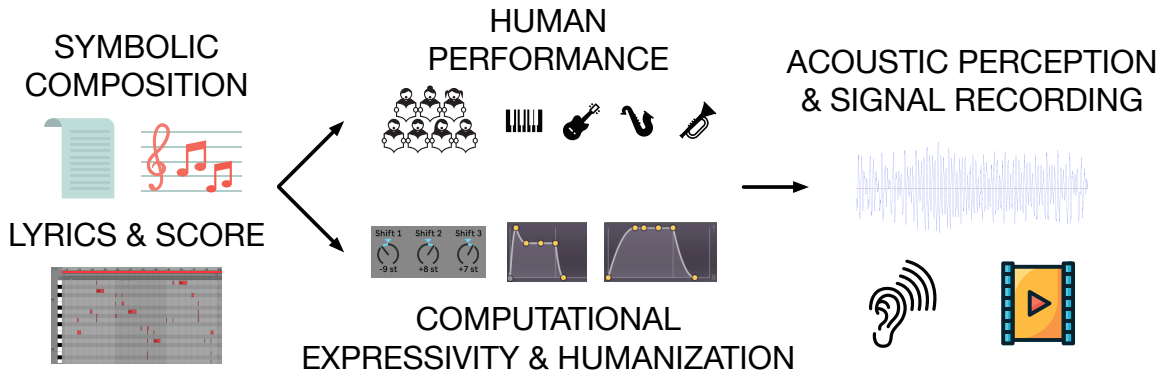

Figure 1: Overview of the main modalities of music production. A music idea is defined in the symbolic domain (score, lyrics). These composition elements are performed by human interpreters or with expressive digital music systems (MIDI effects, virtual instruments, audio processors ...). Acoustic sources or digital rendering blend all elements of composition and expression into an audio signal (audition, digital recording).

the global structure of music and is an essential tool for composition and analysis. Given the needed knowledge of music theory, visual inspection of scores allows direct understanding of the melody relationships, chord progressions, rhythm patterns, repetitions, variations and other fundamental concepts of a music composition. On the other hand, such reduction suppresses the finer details of the actual music performance and acoustic production which are added and interpreted while playing the music piece.

The acoustic information is the most tangible experience of music which blends and conveys all its parameters to the listeners. While the score views instruments as individual and abstract classes, the signal is a particular realisation of the infinite combinations of possibilities existing in the continuous and real world. Thus we have moved from *the* violin in a symbolic layer to *a* given violin instrument, played at a certain point in time and space by an interpreter within an acoustic environment. This yields a dense and complex stimuli to human perception, which mainly propagates through the auditory system and results in the experience of music. In the digital domain, the audio signal is a time series of pseudo-continuous amplitudes (e.g. 32 bits) stored at a high sampling rate (e.g. 44,1kHz) in order to match the frequency range and resolution of the human auditory system. This signal is a highly complex medium of music processing since it implicitly embeds all of its elements along the single dimension of time. Direct and global visualisation in the waveform domain gives little to no cues about the music content, which naturally appears over time by listening while mentally

aggregating the past context and building the understanding for the music continuation. These considerations highlight the critical properties of the musical audio signal:

- ▷ causality as it realises itself forward in time
- ▷ extreme dimensionality along the time axis due to the high sampling rate
- ▷ multi-scale temporal relationships over both the local signal properties (oscillations over hundreds of points), the past musical contexts (several orders of magnitude larger to capture note relationships) and the overall music piece
- ▷ additivity of features (e.g. instruments playing simultaneously) as all the elements of a music piece can be individually perceived while implicitly existing within a single common dimension

From this initial division, the audio information could be thought as the most expressive modality with respect to musical ideas. Yet several intermediate subdivisions and corresponding parameters of expressivity may be interleaved between score and audio. In the next section, we will discuss different layers of music production and their corresponding degrees of freedom allowing to refine musical expression. Other elements and generative functions may be derived around these main modalities of music processing, of which a general taxonomy is proposed in [108].

1.2 Expressivity in Music Composition and Performance

The domain of musical ideas is multi-modal as it can be processed through several data formats, mainly score, text and audio. In addition, the possible outputs can greatly vary in details of interpretation which is the basis of musical expressivity. Depending on such degrees of freedom pertaining to a given piece of composition, a performance can result in various effects (e.g. changing the global tempo or loudness of a piece can make it sound more happy, melancholic or relaxing). To the extent of this introduction, we will analyse different parameters of musical expression in the symbolic domain of score and in the audio rendering. Analogous parameters can be derived from lyrics to detail the mood and emotions conveyed by singing voice, although the research we present is not specifically considering this element of music. Most notably, we specify the discussion to

the domain of computer music analysis and production, nonetheless we intend to draw parallels with expressivity in human performance which would allow to achieve natural sounding audio results and intuitive interactions. The creative potential of computer music should no be restricted to mimicking realistic human performances, yet we can gain insights from analysing those parallels. Comprehensive reviews of computational models of expressive music performance can be found in [286] [148] [149] which broaden the scope of this introduction.

In the past history, symbolic composition of music was carried on *sheets* with notations that are rather agnostic to the instrument of interest. Because of this level of abstraction, the reading of such documents requires an expert knowledge for both composers and interpreters. In order to ease the learning and analysis of music, some novel notations have been developed which offer a more visual representation of the melody and chords. As an example, instrument-specific scores such as the *guitar tablature* which traduces notes into string and fret positions. These visual representations are both simpler to learn for music beginners and convenient for computer processing. Most *Digital Audio Workstations* (DAW, e.g. *Ableton Live, Cubase*) integrate the *piano-roll* as their default representation for music composition (Figure 2), with rows representing pitch classes and columns representing the time grid. The event timings (onset, offset) and other properties (e.g. pitch) are encoded in the *Musical Instrument Digital Interface* (MIDI) standard which allows storing score files and communicating messages between software elements such as *Virtual Studio Technology* (VST) plugins.

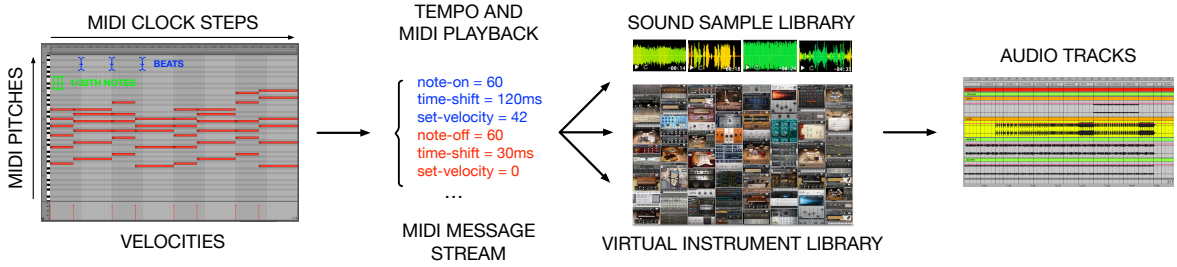

Figure 2: Music production in a Digital Audio Workstation. The composition is defined in the piano-roll representation which is played back according to the master tempo. This generates a stream of MIDI messages to control virtual instruments or trigger samples from a library. This results in audio tracks for each instrument voice that can be down-mixed into an output audio recording.

Thus we start detailing different layers of music expression from the piano-roll representation, and the drum-roll counterpart which represents individual drum tracks as

pitches, as the simplest expression of a music composition. In this quantised representation, a note onset and offset are specified as tick positions which equal to the smallest unit of the time resolution (e.g. number of intervals in a beat, or quarter note). The MIDI note messages also define a pitch value as an integer semitone between 0 and 127 and a velocity (with 0 being silence). Velocity scaling is defined specifically inside each plugin, usually log-scaled to approximately fit the perceptual loudness. This compact encoding of a composition is convenient as it allows visualising and communicating the main content of a piece. However it discards most of the information of an actual performance and a direct rendering of such score would sound unnatural and robotic, no matter the audio fidelity of the output synthesizer. The effect of a realistic performance requires additional modulations and continuous variations that can be appended to the symbolic domain. A pitch bend can be specified as an envelope of cents deviations (percents of a semitone) and an amplitude envelope can be applied to the velocity. Given a sufficient time resolution, expressive timings should as well deviate from the coarse grid as performers naturally move around the exact rhythm. Adding continuous and time-varying parameters to the symbolic representation allows traducing realistic performance features (e.g. a *vibrato*) and conveys much more intentions of musical expression (Figure 3). Yet it leaves out most of the acoustic details and the subsequent rendering system (e.g. virtual instrument) must produce the dense synthesis output given the sparse conditioning of the composition and performance context.

A given melody in the symbolic domain can have many interpretations and another central element of the performance lies in the acoustic details of the instrument. When playing a same note as defined by the pitch and velocity, two different sources (performer plus instrument) will have distinctive features which are embraced in the concept of *timbre* [200] [183]. In the continuous domain, a given combination of pitch and velocity is traduced into a variable spectral energy distribution which can be decomposed into several components, such as harmonics, inharmonics and stochastic coefficients [238] [236]. The perception of these details combines into the concept of timbre and brings an additional layer of music expression [180], for instance instrument acoustics and playing styles. Moreover as timbre carries a sense of acoustic identity, it contributes to the perception the music structure (e.g. recognising the violin and trumpet playing different *voices* in a same song). All elements of the composition and its expressive acoustic rendering are blended in the monophonic (and stereophonic) signal domain, which makes it highly complex and unpractical to manipulate. Akin to a score which is

split into individual instrument voices, multi-track audio is common in both recording settings and production as it allows to separately analyse and process each timbre element before down-mixing to a regular monophonic or stereophonic playback. Other expressive parameters are embed in the acoustic features of music, for instance the spatialisation of the performance (source positions, acoustic environment). Similarly to playing a quantised score without dynamic expression, audio recordings in an anechoic setting sound unnatural and loose part of their musical effect. Multi-track audio also allows to emulate spatialisation, for instance by binaural processing which assigns each source a given angular transfer function fitted to that of the human audition (HRTFs).

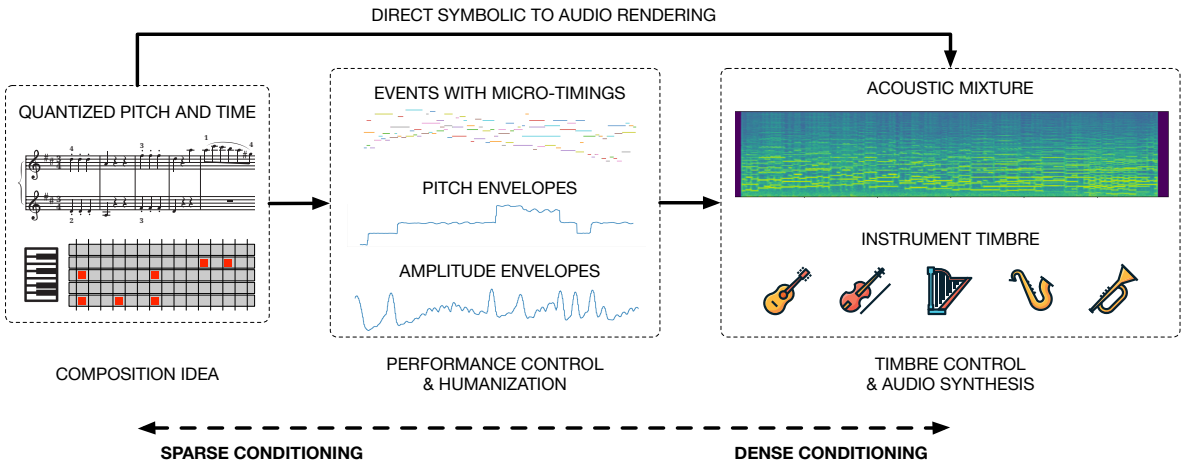

Figure 3: Different layers of music expression from the score (quantised symbolic input) to the performance controls (fine-grained and dynamical) and ultimately the acoustic rendering (instrument timbre, room acoustics). Some systems perform direct score to audio synthesis and implicitly process the performance features.

1.3 Problem setting

In this introduction, computer music production is mainly summarised as the process of going from a compressed symbolic representation of the composition idea to the dense audio mixture that incorporates all elements of performance and acoustic rendering. Within this process, the thesis research focuses on the synthesis task that is to generate and model the acoustic variability pertaining to the given musical context. This context may be explicitly defined, for instance by sparse high-level targets such as pitch, velocity and instrument classes. Or it may be implicitly carried by another low-level acoustic example, for instance the performance recording of an instrument as the basis of the

synthesis of another one, so that the task involves analysis and synthesis. In both cases, we are primarily interested in the capacity to represent and control variations in timbre via meaningful parameters that would allow musical interactions in the acoustic domain. These direct interactions in the acoustic domain aim at modelling perceptual properties which are discarded from the symbolic representation of music, yet playing a major role in the production of music and its semantic effects. While a human performer naturally interacts within the continuous realm and learns to integrate the different elements of interpretation and acoustic expression by practice, this raises challenges for computational models of music production.

One direct approach to computer music production is the sample-based rendering that relies on a library of pre-recorded individual audio samples that are retrieved according to the event targets, for instance the pitch, velocity and instrument classes of notes in a melody. In this case, the audio fidelity and control accuracy are ensured by the use of annotated pre-recorded audio instead of a synthesis engine. However, the expressivity of sample-based rendering is limited because it concatenates individual events which are selected by categorical attribute targets. For that reason it does not produce a natural articulation which requires to model the relationships in a series of notes and its controllability is limited as it does not offer continuous controls on the acoustic variations belonging to the given event classes. Since the sample selection relies on the library metadata, the approach has a limited representation of its semantic content and besides the coarse categories such as pitches, dynamics and instruments it does not meaningfully traduce finer timbre details. For that reason, browsing through large sampler libraries is often tedious and arbitrary.

Another common approach to music production relies on Digital Signal Processing (DSP) to implement synthesizers with continuous controls over the generated signal properties [211]. It should be noted that DSP can be combined with samplers, for instance as an audio effect applied to the sample play-back in order to control or emulate certain acoustic properties (e.g. reverberation). The synthesizer controls are related to the underlying signal processing, for that reason there is often a gap between their functions (e.g. rate of a modulation, frequency of a filter) and the perceptual variations they produce. Modern digital synthesizers generate audio of high quality and rich diversity with real-time and continuous controls, however this is often achieved by processing pipelines of increased complexity that feature a large amount of parameters.

In that sense, we often observe a trade-off between the capacity of the synthesizer to produce diverse sounds and the intuitivity of its controls [130]. The synthesizers can be enhanced by meta-controls which govern multiple parameters and yield a more compact interface, nonetheless they require extensive practice and time to empirically tune their parameters until reaching the desired sound. Synthesizer configurations are often saved as presets with metadata in order to browse many possible settings and explore different templates of sounds that can be produced. As with audio sample libraries, the browsing of the synthesizer presets relies on metadata which only partly represent the underlying acoustic and semantic contents. In order to generate a rich output, the synthesizers often feature non-linear DSP operations (e.g. frequency modulation) so that parameter changes can have radical and unpredictable effects. This hinders the use of presets as a basis to craft other sounds which may be perceptually similar but may correspond to much dissimilar parameter configurations.

From this perspective, we observe that digital synthesis has the potential to generate all the acoustic variations we may target but faces the challenge of providing intuitive controls and interactions with semantic properties. Analysis and synthesis techniques allow the extraction of audio parameters which can be manipulated in order to generate new samples. Such audio representation is more interpretable since it can be directly related to acoustic observations rather than to arbitrary metadata which somehow describe the parameter configuration. However accurate sound models for analysis and synthesis often have a high number of processing parameters and representation dimensions which are impractical for musical interactions. Although the representation is grounded in the acoustic domain, it hardly traduces semantic properties which may not have a direct inversion mechanism. For instance, the perception of timbre has been studied based on human ratings to asses the degree of dissimilarity between the sounds produced by different instruments. The distribution of these ratings can be organized into timbre spaces [98] of low dimensionality that traduce the perceptual relationships of musical sounds as continuous distances. These representations correlate with acoustic perception [68], however they are not invertible and there is no synthesis model with parameters corresponding to the analysis dimensions.

In this thesis, we apply deep generative models to the task of musical audio synthesis to tackle the challenges of large-scale and meaningful interactions with audio sample libraries by learning probabilistic representations of the data. We focus on in-

vertible representations for analysis and synthesis to construct generative spaces of low dimensionality that model the underlying structure of an audio library. In the unsupervised setting that does not rely on metadata, we use probabilistic regularisations as a mean to automatically organize the observed acoustic variations into a smooth and compact space which allows visualisation, scattering the data on the learned analysis dimensions, and generation by using these continuous dimensions as synthesis parameters. The assumption of this probabilistic approach is that audio data with consistent semantic properties occupy a sub-space of the observation space, a manifold [13] of lower dimensionality that relates to the underlying nature of the data (e.g. physical constraints, generative factors). In our case, these dimensions are those of acoustic perception and temporal structures which correspond to meaningful musical audio that we observe in a library and learn to represent by analysis and synthesis. We extend this approach by relying on metadata provided with sample libraries to disentangle specific factors of variation in the learned representations. These additional explicit learning constraints enable more predictable transformations and control over the generation of specific musical attributes. It can be implemented by separating the data in domains (e.g. different instruments) which we learn to map by disentangling their specific features from a domain invariant representation. This shared representation relates to higher-level music properties, for instance symbolic abstractions (e.g. pitch) or acoustic variations that are found in the joint distribution of multiple timbres. It applies to timbre transfer and audio stylisation by analysing a source sample and imposing it part of the auditory features learned from another target instrument domain. In the semi-supervised setting, we use metadata to learn specific control dimensions for target attributes while letting unsupervised dimensions model the remaining acoustic variations pertaining to the given conditions. Such representation allows controllable sampling of new audio which is consistent with the library and its metadata, including the recombination and inversion of multiple perceptual attributes.

While we emphasize the representation learning and interaction properties that we target by probabilistic generative modelling, a significant research is carried into developing neural networks adapted to processing audio and music information. This raises multiple challenges due to the high dimensionality of audio time series which require multi-scale representations of both local and long-term structures, as well as computationally efficient models for low latency synthesis and fast training on constrained amounts of data.

1.4 Thesis organisation and main contributions

In the second section, we introduce some fundamental concepts of audio and music processing with an emphasis on classical models. We review the common representations of the audio waveform in the time domain and in the spectro-temporal domain by short-term analysis. Based on the different acoustic representations of audio, we introduce methods for decomposition and feature extraction which enable higher-level analysis tasks for understanding the music structure. The music information retrieval setting is presented as an inference mechanism from low-level acoustic features that are processed and reduced to target music properties in the symbolic domain. We as well review the main techniques for audio synthesis as the inverse generation mechanism of the dense acoustic rendering given sparse conditions. We observe that these two processes are often disjoint, meaning that most of the features and transformations used in information retrieval are not invertible. For that reason we detail methods of analysis and synthesis which are of great interest regarding this thesis research but highlight the challenge of modelling invertible audio representations which are perceptually relevant and intuitive to control. We conclude by a generic introduction of supervised machine learning for audio and music processing, as throughout the thesis we intend to build on representation learning as an analysis and synthesis tool.

The third section focuses on generative modelling with deep neural networks which strongly relies on unsupervised learning and probabilistic models. In contrast with supervised learning which aims at retrieving ground-truth labels of the data, generative modelling is posed as a problem of estimating the continuous data distribution underlying a finite dataset of observations. As this problem cannot be evaluated in closed form, we review several frameworks which use neural networks to parameterise a family of approximate distributions that is optimised to fit the observed data. The introduction of these fundamental models for generative modelling is organised with respect to their information structures, notably by the definition of a prior distribution over a latent representation and the ability to perform analysis and synthesis. We as well categorise their different optimization objectives which are usually defined by some explicit likelihood or by some implicit distribution matching, either in the data space or as constraints over the latent representation. Given this review of unsupervised learning models, we detail some specific generative applications which incorporate conditioning to allow a semi-supervised control as well as domain translations. We conclude by pre-

senting different kinds of metrics for evaluation, some of which are closely related to the training objective of distribution matching whereas others emphasise data-dependent properties or accuracy with respect to control and perception.

The application of generative models in audio synthesis is presented in the fourth section. We review generic models of neural audio synthesis and spectrogram inversion before focusing on music signal generation. In that case, control interactions and the processing of timbre are central properties of the models that we divide between implicit and explicit representation learning of timbre. These models focus on fine-grained acoustic modelling and in the later we detail methods for end-to-end learning of analysis and synthesis sound models, for instance using auto-encoders as done throughout a significant part of this thesis research. Lastly, we present models focusing on score to audio synthesis as well as perceptual embeddings which can be used to enhance the training of these different kinds of generative audio models.

In the fifth section we present some of the work done and published throughout this thesis, which references were mainly detailed in the sections three and four. Our contributions in the field of implicit timbre models have relied on domain-translation techniques which we specify to unsupervised timbre transfer. Since these methods require adversarial training and do not scale well to many domains, we propose to apply non-parametric distribution matching to audio spectrograms which offers a stable and scalable translation objective. We perform many-to-many timbre transfer in a single neural network by relying on an expressive conditioning mechanism to apply domain-specific statistics to hidden features. In addition to statistical evaluations, we propose to visualise the topology of acoustic descriptors and show that our method effectively converts audio domains which is traduced by implicitly matching the generated distributions of acoustic descriptors with that of the target. Our contributions in the field of learning representations of timbre are based on analysis and synthesis with auto-encoders so that dimensions in the latent representation could structure variations in timbre which can be manipulated and inverted to audio. We show that by applying an adversarial regularisation to the latent representation we can disentangle attributes of note samples that are independently controlled, for instance pitch and instrument classes or playing styles. We pre-train corresponding classifiers and assess the accuracy of the attribute controls which can be used for intuitive audio sampling with custom targets. We complement this controllable spectrogram generation model with an inversion

model which all-together allow fast waveform generation in a single pass. Our further contributions have tackled the challenging problem of end-to-end waveform modelling, which requires efficient learning of both local properties and longer-term dependencies that span many orders of magnitude. To this extent, we propose a hierarchical model for granular sound processing which learns a local acoustic representation as well as a structured temporal embedding of short-term waveform features. We demonstrate the efficiency of the waveform learning with our proposed short-term noise filtering and overlap-add approach against a convolutional neural network baseline. The resulting model is light-weight and can run in a prototype virtual drum-machine. The audio sample generation can be performed as interpolation in the acoustic embedding or conditional sampling in the temporal embedding. By performing interpolations in the temporal embedding, we show that we can morph audio samples. Based on this approach to short-term waveform modelling, we propose a second method to structure the learned acoustic representation and interact with it. Instead of learning a temporal embedding, we propose to apply discrete representation learning as a mean to decompose the acoustic representation into a finite set of latent features of timbre. By doing so, we show that we can perform timbre transfer directly in the waveform domain and from arbitrary sources including the possibility of voice-driven synthesis. Moreover, we can analyse and map the learned latent features with acoustic descriptors and control the synthesis by explicitly defining some target acoustic variations. As part of the evaluation of our proposed models, we discuss both the learning performance as well as qualities related to control and usability. Amongst major major challenges in the dissemination of machine learning tools are the needed data and computation resources to train and deploy models. Thus we conclude our presentation with a group research carried on compressing neural networks across a broad range of tasks including music information retrieval and audio synthesis, which demonstrates the possibility of maintaining the model performances while drastically reducing their sizes. These results are significant steps towards real-time capable neural audio processing models running on laptop CPU or embedded hardware.

In the last section, these results and academic publications are summarized along with several related projects carried in the course of the thesis. My topic was motivated by both a strong interest in science and audio technologies as well as a passion for music and generative arts, which I could intersect during such projects. Finally, future works are discussed for modular score to audio generation with expressive performance control.

2 Audio and Music Processing

2.1 Audio Representations

The digital waveform. The audio domain encompasses a great variety of physical phenomenons such as speech production, music sounds and environmental noises which result from the emission of a continuous pressure signal. The most general counterpart to the acoustic oscillation is the digital waveform (Pulse-Code Modulation), a unidimensional time series which approximates any sound in the audible range if sampled at a sufficient rate (e.g. 44,1kHz). It is also the least lossy storage and play-back of audio information as both high sampling rates and high floating point precisions can approximate the continuous time and real-valued amplitude of the acoustic signal. In that sense, there is a trade-off between fidelity and dimensionality which is inherently related to digitisation: the resolution of the time axis sets the higher frequency bound according to the Nyquist law and the amplitude resolution (audio bit depth) sets the dynamic range and the noise level due to quantisation errors. The setting of each resolution can be optimised by choosing a sampling-rate adapted to the highest frequency expected in the audio (sampling at twice the maximum frequency) or by using an amplitude-dependant quantisation scale (e.g. μ-law). The later applies a non-linear transformation to the amplitude with a log-shape that follows the dynamic perception of loudness in the human audition and allows to mitigate quantisation artefacts down to a 8-bit depth. Given $x \in [-1, 1]$ a linear waveform amplitude and $\mu = 255$, the 8-bit transformation is:

$$F(x) \equiv sgn(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}. \tag{1}$$

A certain compression can be achieved, nonetheless the audio waveform remains a highly dimensional and impractical signal to visualise. Moreover, the local properties of the waveform (e.g. sensitivity to time shifts) do not align with the macroscopic perception of sound such as invariance to phase shifts and grouping of slowly varying oscillations.

Short-term representations. Another common class of audio representations are based on spectro-temporal transformations of the waveform into 2-dimensional features. Using a sliding window (possibly with overlap), the time axis is down-sampled into frame series of spectral coefficients such as those of the Discrete Fourier Transform

(DFT) that yields the Short-Term Fourier Transform (STFT). Formally, we denote the real-valued waveform time-series as $\mathbf{x} = \{x_0, \ldots, x_{L-1}\}$ and a chosen window function $\mathbf{w} = \{w_0, \ldots, w_{N-1}\}$ applied with a fixed hope size $H \in \mathbb{N}$. The 2-dimensional STFT complex spectrum $\mathbf{X}(m, k)$ is computed as:

$$\begin{aligned} \mathbf{X}(m,k) &\equiv \sum_{n=0}^{N-1} \mathbf{x}(n+mH)\mathbf{w}(n) \exp^{-2\pi ikn/N} \equiv |\mathbf{X}(m,k)| \exp^{i\Phi(\mathbf{X}(m,k))} \\ \exp^{-2\pi ikn/N} &\equiv \cos(2\pi kn/N) - i\sin(2\pi kn/N) \text{ (the orthogonal Fourier basis)} \end{aligned} \tag{2}$$

with $m \in [0 : M]$ and $k \in [0 : K]$. The down-sampled time-frame axis is of length $M = (L-N)/H$ and the frequency axis is of size $K = N/2$ with its range set according to Nyquist law so that $f_K = f_s/2$ (f_K being the frequency of the K-th spectrum bin and f_s the audio sampling rate).

The extraction of stationary frame features (invariant to small translations along time) better traduces the auditory perception of slowly varying oscillations which appear as stable energy components in the spectro-temporal plane. The complex valued STFT can be factorised into magnitude $|\mathbf{X}(m, k)|$ and phase $\Phi(\mathbf{X}(m, k))$ real-valued components. The visualisation of spectro-temporal representations, often in the form of magnitude spectrograms, is eased akin to reading an image projection of the audio signal. Yet it must be noted that spectro-temporal audio properties differ from images and should be processed adequately:

- ▷ the 2 dimensions are not symmetrical as the spatial dimensions of an image, in particular the time-frame axis is translation invariant but shifts in the frequency axis modify the sound pitch
- ▷ spectrogram "pixels" can be shared by multiple sources due to the additivity of sounds whereas image pixels mostly belong to a single object (occlusion of the background)
- ▷ sound objects are non-locally distributed as spectral energies of a given sound can be far apart (e.g. harmonic ratios) whereas neighbouring pixels mostly belong to a same visual object

To a certain extent, the frequency dimension may be seen as "colour channels" which can be treated as independent time series, although this assumption has several flaws:

energy often leaks across several frequency bins and a given energy component varies over-time and is likely to jump over channels. Methods for re-scaling the linear frequency axis $f_k = k * f_s/N$ have been introduced and alleviate some of these issues by using a non-linear frequency bin distribution which aligns with human perception and better traduces some specific sound properties. An example of such is applying the Mel-scale which was empirically set to follow the quasi-logarithmic human perception of pitches as $m = 2595 \log_{10}(1 + f/700)$. The use of a logarithmic bin distribution along the frequency expands the resolution in the low frequencies and compresses the higher end of the spectrum. This property is as well often desirable with respect to pitched musical sounds which are prominently composed of harmonic energy components. Because these components follow a frequency distribution such that $f_i = i * f_0$, with f_0 the fundamental frequency (pitch), it directly implies that harmonics of low pitched sounds will be closer to each other than harmonics of high pitched sounds. Thus, for a given size of the frequency axis, the separation of pitched musical sounds is improved by using a logarithmic frequency spectrogram representation. To this extent, musically informed spectrogram representations have been specifically proposed, of which the most common is the Constant-Q Transform (CQT [26]). In the western tonal standard, musical notes are organised in semitone steps and grouped in octaves of twelve pitches (well-tempered scale). Because an octave jump corresponds to doubling the frequency of a pitch, the distribution of musical notes is logarithmic and it is the scale adopted by the CQT. However, using a logarithmic scale raises issues of amplitude normalisation which have motivated the use of a constant-Q transformation in the sense of keeping a constant $Q = f_k/\Delta f_k$ in reference to the *Quality* factor of a filter in signal processing given a bin center frequency f_k and a width Δf_k. As a result, the CQT frequency axis is equivariant with respect to pitches and octaves, which means that harmonic distributions are translation invariant. In the context of pitched sounds, the CQT spectrogram shares many of the properties of images and offers a convenient musical audio representation (Figure 4).

The previous audio representations are derived from the magnitude of the complex-valued STFT. Another widely used transform is the real-valued Discrete Cosine Transform (DCT) which is usually computed as:

$$\mathbf{X}_{DCT}(k) = \sum_{n=0}^{N-1} \mathbf{x}(n) \cos\left[\frac{\pi}{N}(n + \frac{1}{2})k\right] \tag{3}$$

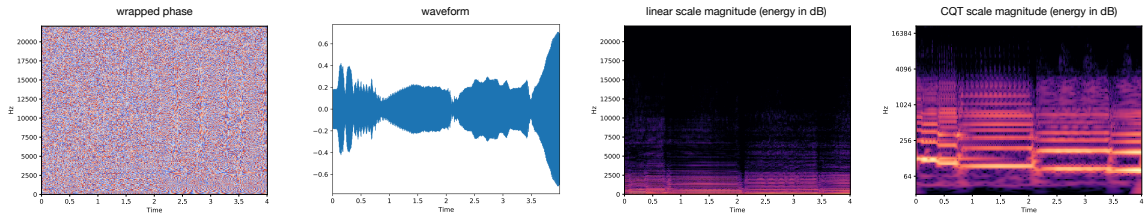

Figure 4: Visualisation of a 4-second excerpt of Cello solo. Different spectro-temporal representations are derived from the raw waveform, the magnitude spectrograms allow direct observation of the different notes played with a clear temporal segmentation. The CQT has an improved separation of the note pitches and their corresponding harmonic distributions.

for a signal frame of size N. The DCT is used in signal processing [216] for compression and in audio for transient modelling [275]. It can as well be combined with the STFT magnitude spectrogram to compute the coefficients of the Mel Cepstrum (MFCC) by applying the DCT to the log-magnitude Mel-spectrogram. While the STFT decomposition extracts periodic components from the time domain signal, applying the DCT subsequently extracts periodic components in the spectrum and yields a smoothed amplitude envelope of the spectrogram. It should be noted that both the complex-valued STFT and the DCT have an exact inverse transformation and thus allow analysis and synthesis (retrieving the input signal) but the non-linear frequency representation derived from the magnitude are not. Although convenient for visualisation and analysis, none of the representations such as Mel-spectrogram, CQT and MFCC have a straightforward application to audio synthesis. The Non-Stationary Gabor Transform (NSGT) allows an adaptive signal representation along both the time and the frequency dimensions while remaining invertible. This transformation is based on the Gabor signal expansion into a periodic sum of elementary signals (grains) and can be thought as a sampled STFT with windows of different sizes. An application is the Constant-Q NSGT [273] which provides a frequency scaling that is equivariant with respect to pitches and octaves while satisfying the condition of invertibility.

2.2 Feature Analysis and Visualisation

The aforementioned magnitude representations facilitate the visualisation of the spectro-temporal patterns of a given waveform but have discarded the phase information from the complex-valued short-term spectrum. This is due to the randomness of the raw

wrapped phase which does not exhibit distinctive features such as those of the energy amplitude (Figure 4). Interpretable visualisations of the phase information require additional processing steps [295] such as unwrapping of the angular phase, computing time derivatives (instantaneous frequency) and frequency derivatives (group delay). Since most audio features are derived from the spectral energy distribution, we do not detail the processing of phase information. The magnitude spectrogram is a lossy representation of the digital waveform (e.g. halved dimensionality), nonetheless it provides a rather faithful description of the broadband acoustic information and is the basis for several analysis processes which we categorise as decompositions or extractions.

Feature decomposition. The decomposition of an audio representation aims at separating different components of a given sound (Figure 5), which can be re-assembled in order to retrieve the original representation. This is the basis of what will be referred as *analysis/synthesis* methods. Moreover, a given decomposition can serve as a pre-processing step for a subsequent analysis method (Figure 6). Generally speaking, audio data are often mixtures of sound sources which overlap in time and frequencies, for instance an ensemble of instruments playing a given piece of music along with some background noises. Ideally, the first processing step of an audio signal analysis would be to decompose the mixture in individual tracks for each instrument. Source separation has extensively been studied [196] [214] and it remains an open challenge which is beyond the scope of our discussion (e.g. the number and type of sources may not be known in advance). To the extent of music production, the results of source separation tend to leak in between instruments and the audio quality cannot match that of real multi-track recordings (e.g. part of the timbre is suppressed, signal-to-noise ratio is low). Considering the sounds produced by a given instrument, these may be coarsely classified as either harmonic (the sustain of a note) or percussive (the attack of a note). This classification has a dual representation in time and frequency domains, impulsive components of the waveform appear as noise-like spectrum distributions (broad bands of energy) whereas slowly-varying waveform components yield sharp spectrum distributions (peaks of energy). This property has been efficiently used for harmonic-percussive decomposition in the spectrogram domain with median filtering [79]. The smoothing effect of a median filter is either applied along the time dimension to remove transients

in $\tilde{\mathbf{H}}$ or along the frequency dimension to remove harmonics in $\tilde{\mathbf{P}}$:

$$\begin{aligned}\tilde{\mathbf{H}}(\mathbf{m},\mathbf{k}) &= median(|\mathbf{X}(m-l_h,k)|,\ldots,|\mathbf{X}(m+l_h,k)|)\\ \tilde{\mathbf{P}}(\mathbf{m},\mathbf{k}) &= median(|\mathbf{X}(m,k-l_p)|,\ldots,|\mathbf{X}(m,k+l_p)|)\end{aligned} \tag{4}$$

with $2*l_h+1$ and $2*l_p+1$ the filter sizes along each dimension. The harmonic or percussive enhanced magnitudes are compared and converted into binary masks applied to the original spectrogram. This yields a decomposition such that $|\mathbf{X}| = \mathbf{H}+\mathbf{P}$. It should be noted that the aforementioned decomposition method is not restricted to harmonic distributions but rather to narrowband energy components which vary slowly over time (without constraint such as the integer ratio of frequencies).

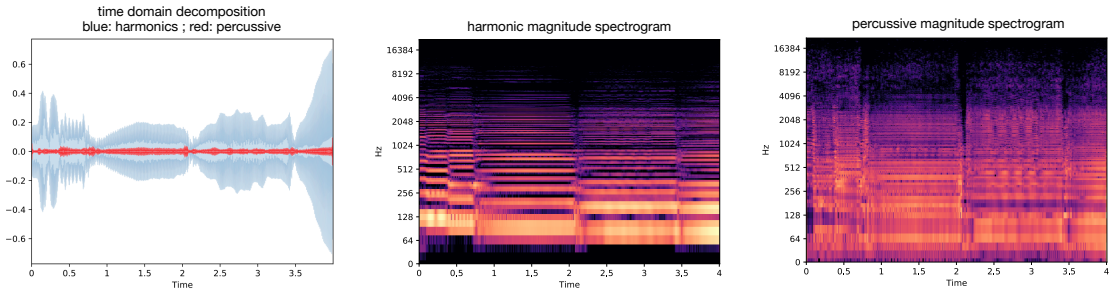

Figure 5: A 4-second excerpt of Cello solo is decomposed into harmonic and percussive components by median filtering in the spectrogram domain (visualisation in log-frequency). The corresponding decomposition is shown in the waveform domain.

Feature extraction. As displayed in figure 5, the spectrogram decomposition emphasises the underlying content of an audio such that the different notes of a melody are clearly visible in the harmonic component. However, the interpretation of the percussive component is not straight-forward as some patterns remain over time, which motivated the use of a finer decomposition in [62]. Such decompositions enhance the visualisation of sound, yet they do not estimate the parameters of the underlying content which have to be extracted in a subsequent process (e.g. pitches, onset times). Since both the input and output representations are the same, the dimensionality of the decomposition is multiplied by the number of output components. On the other hand, extraction methods aim at analysing specific features in the input and reducing it to a set of target properties (while loosing the capability to directly retrieve the input from the composition of simpler outputs). The relevance of the audio feature

extraction highly depends on its correlation to human perception or to the subsequent tasks it allows (yielding an analysis space) and the compression of information directly traduces the multi-scale nature of audio signals. Following 2.1, we coarsely categorise the features of music signals from the low-level waveform amplitude to the short-term intermediate-level (frame-wise) and towards the longer-term semantic properties of music (Figure 6). Many frame-wise descriptors have been proposed to traduce the dynamic acoustic properties of a sound, amongst which spectral descriptors computed on spectrogram frames [203]. In comparison with spectral transformations which map to a specific frequency scale (e.g. Mels), these descriptors are scalars that summarise the overall spectral distribution of a given frame. For instance the Spectral Centroid (SC) is a weighted mean of the bin frequencies multiplied by their magnitudes:

$$SC(m) = \frac{\sum_{k=0}^{K} f_k * |\mathbf{X}(m,k)|}{\sum_{k=0}^{K} |\mathbf{X}(m,k)|} \tag{5}$$

which correlates to the feeling of *brightness* of a sound [99]. As introduced in 1.2 the perception of musical sounds may be divided into pitch, loudness and timbre which have corresponding audio descriptors. While loudness has standardised measures (e.g. A-weighting of the power spectrum), the estimation of pitch is not straight-forward (e.g. cepstrum analysis [178]) and acoustic descriptors of timbre could only be evaluated with respect to their degree of correlation to human ratings (e.g. instrument similarity ratings [181]). As noted in [242], there may be a divide in the use of acoustic descriptors for research in timbre perception and music information retrieval which has crafted additional task-specific features (e.g. for onset detection and transcription). Nonetheless, these different feature extractions are a base for higher-level information retrieval and semantic analysis. While short-term features have a fixed distribution in time given the spectral analysis constant frame-rate (yielding feature envelopes such as pitch contour), the higher-level music properties span longer-term and variable-length temporal contexts. For instance, musical notes can have different durations of the order of seconds whereas the spectrogram hop size have a fixed duration of the order of tens to hundreds of milliseconds. This implies that modelling the higher-level music structure and perception requires further temporal down-sampling, segmentation (e.g. event onsets and offsets) and classification into categorical labels (e.g. event pitch, velocity and instrument classes). These tasks are referred as Music Information Retrieval (MIR [188]) and the extraction of an explicit music structure (e.g. piano-roll) serves

the prediction of global music attributes such as the key, the tempo, the time signature, the rhythmic patterns and chord progressions. Both symbolic structure and acoustic features contribute to the perception of music, the fusion of these modalities ultimately enable the analysis of orchestration which is the art of creating musical effects at the interplay of composition and instrumentation. Such effects take into consideration the tessitura (pitch range), playing modes and timbre relations in order to distribute instrument voices and achieve the desired musical sound mixtures.

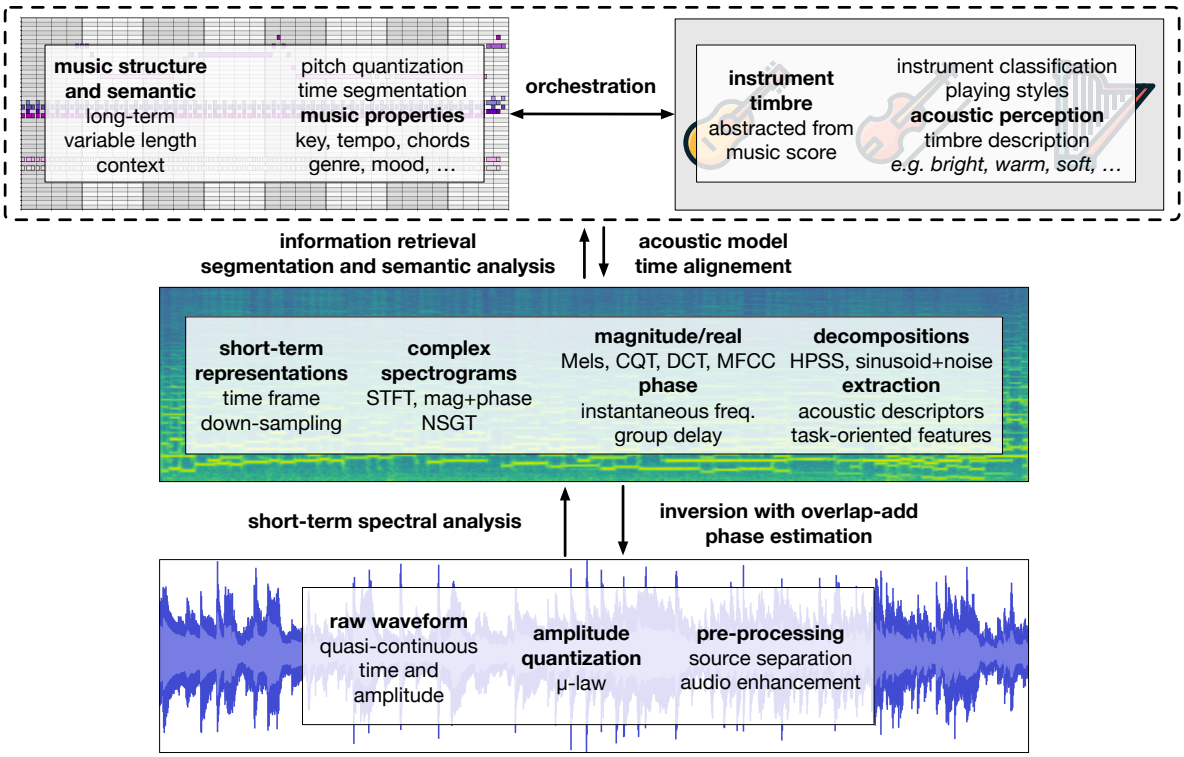

Figure 6: The multi-scale nature of audio and music information is divided into three levels of representation. The low-level waveform is first down-sampled by short-term analysis which extracts local spectro-temporal features. Temporal segmentation and classification allow higher-level information retrieval and semantic extraction of music and timbre relationships.

2.3 Audio Synthesis

Feature analysis has been detailed as a bottom to top-level inductive process which leads to musical abstractions such as score properties. Formally, a parametric analysis model $\mathcal{F}_\phi$ takes an input sample $\mathbf{x}$ (e.g. an audio sample, a spectrogram) and extracts some higher-level properties $\mathbf{y} = \mathcal{F}_\phi(\mathbf{x})$ (e.g. a label). Its counterpart is the

top to bottom-level generative process $\mathbf{x} = \mathcal{F}_\theta(\mathbf{y})$ which ultimately results in audio synthesis of the target timbre under the composition and performance context (e.g. articulation of a melody, playing styles). Digital audio synthesis systems [125] [261] can be broadly grouped into three categories: physical models, corpus-based methods and signal processing techniques (which will be further detailed as either abstract or analysis/synthesis). Each of these audio synthesis system types are mostly parametric models which differ in the meaning of their parameters θ and the underlying structure of the synthesis parameter space that leads to signal generation. As discussed in the introduction, the relevance of an audio synthesis greatly depends on the control qualities and thus parts of the synthesis parameters θ^i usually serve as inputs and controls to the system whereas the remaining model parameters θ^m define the internal capacity and state of the system. These controls can allow direct user manipulations (e.g. turning a knob, pressing a key) and with respect to the top to bottom information processing shown in Figure 6, the higher-level context $\mathbf{y}$ (e.g. melody, pitch contour) should interface with the synthesizer input parameters in order to render the target music properties in the signal domain. As a result, we can generally define the generative process of a conditional synthesis model as:

$$\mathbf{x} = \mathcal{F}_{\theta^m}(\mathbf{y}, \theta^i). \tag{6}$$

Physical simulation. To this extent, physical models [245] [16] simulate the natural phenomena of acoustic production (e.g. modes of vibration, material interactions) for a given source. The internal parameters are usually related to the simulation technique (e.g. wave-guides, modal decomposition, finite elements) and the input parameters traduce physical variables such as shapes (e.g. length of a string, instrument body), materials (e.g. stiffness, density) and excitation (e.g. bowing speed, hit intensity). Part of the input parameters may be found by physical identification from a given source (shapes and materials) and the other excitation parameters are controls which emulate the playing of the instrument. As such, physical models benefit from a direct interpretability with respect to what they simulate (e.g. changing the length of a string changes its pitch) and can achieve highly realistic audio. However, they have a limited expressiveness since they only render a certain physical phenomenon, every sources and playing modes require a dedicated model and simulation. In addition, the simulation complexity can quickly increase along with the physical phenomenon complexity (e.g.

non-linear interactions) and required accuracy (e.g. resolution of a mesh, numerical stability constrains).

Corpus-based methods. Instead of performing the synthesis, corpus-based methods use libraries of audio recordings as source material for audio generation, for instance a note sampler that plays-back a certain audio sample corresponding to some control targets (e.g. instrument, pitch, velocity). A whole performance can be assembled by concatenating individual event samples corresponding to the notes of a melody, yet such sampler does not offer expressive controls over the timbre (e.g. continuous acoustic variations for a given pitch and velocity) and the articulation (relationship between notes). Some samplers and wavetables include modulations and transformations (e.g. audio effects) to manipulate the sample play-back. This allows a certain degree of expressivity but does not scale to large corpuses as a given hand-tuned play-back setting may only apply the desired effect to a certain sample. Another class of corpus-based methods does not rely on event-level audio samples (e.g. with an ADSR amplitude shape of a few seconds) but rather uses shorter signal windows (e.g. around 100ms.) as independent templates (without inherent time structure) that can be combined to form longer and more complex signals. These signal templates can be extracted by slicing all audio samples of a given library into fixed-length *grains* which are the basis of granular sound synthesis [225]. Because the duration of audio grains (e.g. 100ms) can be several orders of magnitude lower than usual audio recordings, this quickly yields a large number of audio templates to manipulate. To this extent, a grain library is analysed with audio descriptors which assess the acoustic relationships across grains that are projected onto a reduced feature space (one dimension per descriptor as shown in Figure 7). This allows visualisation as well as smoothed audio concatenation (given a sufficient grain density) as close neighbours should be matched acoustically. The issue of phase misalignment at the grain edges can be circumvented by overlap-add with symmetrical windows to fade-in/out the grain edges [84]. Usually, the analysis and projection into the grain space does not take into account the temporal relationships between grains which are solely compared in terms of acoustic similarity. Accordingly, synthesis paths in the grain space often have a good sounding texture but do not exhibit meaningful temporal structures. Granular and concatenative methods [233] have also been applied to re-synthesis [231] of a given source sample which conveys its intrinsic structure. In this approach, the source sample is sliced into short segments which are

matched with their corresponding grains in the library. The ordered input series of features can be recomposed with those of a given sound library according to a certain matching criterion based on distances in acoustic descriptors.

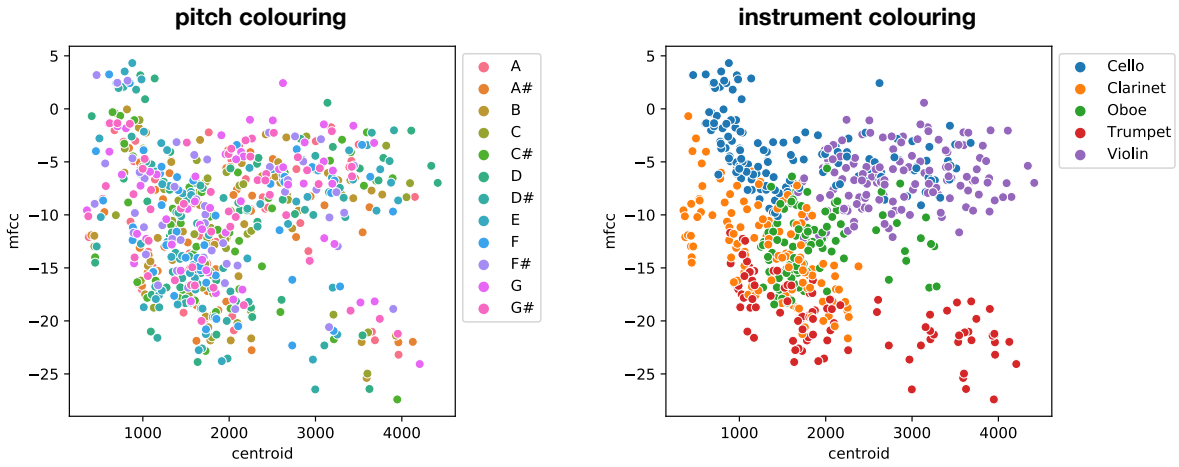

Figure 7: An analysis of individual notes across the 12 pitch classes and 5 instruments (full tessitura). Two descriptors are chosen as dimensions for the scattering space, the Spectral Centroid and the mean of the 20 first MFCCs which are extracted for each sample. On the right plot with instrument colouring, we can observe some timbre grouping with different regions pertaining mainly to certain instruments (e.g. top right for the violin).

Signal models and analysis/synthesis. Lastly we will detail methods based on digital signal processing (DSP) operations which can either be abstract or based on spectral models. In the abstract approach, the generic DSP parameters define a certain state of the system from which the signal is generated, without explicit definition of the target features. For instance non-linear operations such as frequency modulation (FM synthesis [45]) and wave-shaping which can create complex timbres with only a few parameters, although these parameters are hard to interpret and thus little predictable in terms of output sound and musical context. On the other hand, spectral models often rely on an analysis/synthesis framework [244] and an underlying sound model. The analysis stage extracts audio parameters $\theta^i = \mathcal{F}_\phi(\mathbf{x})$ from the signal, these features correspond to a certain spectral representation and are interpretable parameters for a subsequent synthesis mechanism $\mathbf{x} = \mathcal{F}_{\theta^m}(\theta^i)$ which inverts spectral features back to the signal domain. Given some context $\mathbf{y}$, the synthesis parameters can be extracted from a given signal, manipulated according to a certain transformation model $\hat{\theta}^i = \mathcal{T}(\theta^i, \mathbf{y})$ and synthesised into a new signal $\hat{\mathbf{x}} = \mathcal{F}_{\theta^m}(\hat{\theta}^i)$ (Figure 8). The spectral model

is often a basis for decomposing a complex acoustic information into simpler short-term features that can be transformed and recombined in order to achieve another complex target sound. The complex STFT allows decomposing any signal into a sum of sinusoids with variable amplitudes and phases at linearly distributed frequencies and the inverse STFT (iSTFT) allows overlap-add reconstruction [126]. As detailed in 2.1, the phase is often discarded from the processing of the spectro-temporal representation which prevents direct inversion of the magnitude spectrogram with the iSTFT. Phase estimation from the magnitude spectrogram can be performed with the standard Griffin-Lim algorithm (GLA) [100], an initial random phase is appended to the magnitude and iteratively refined by successively applying STFT $\circ$ iSTFT which only updates the phase until convergence. The reconstruction quality can be increased by using several hundreds of iterations which can cause a significant latency, yet noticeable artefacts remain. The STFT spectrogram representation does not assume a specific sound model as it provides a complete orthogonal basis on which any signal can be decomposed. Generally speaking, invertible representations such as the Fourier transform are bound to preserve (or increase) the input dimensionality and are often over-parameterised with respect to specific subsets of sounds. For instance, a harmonic sound will have a sparse energy distribution which is concentrated over a set of spectral peaks, leaving the other frequency bins unused. Additive synthesis relies on an oscillator bank which simulates the spectral peaks of a sound, yielding a more compact representation as the amplitude, frequency and phase information only need to be processed for the most prominent energy components. These tuned oscillators are summed over time in the signal domain as:

$$\begin{aligned} \mathbf{x}(n) &= \sum_{k=1}^{K} A_k(n) \sin(\Phi_k(n)) \\ \Phi_k(n) &= 2\pi \sum_{m=0}^{n} f_k(m) + \Phi_{k,0} \end{aligned} \tag{7}$$

for $n \in [0 : L-1]$ the signal length parameterised with K oscillators of time-varying amplitudes A_k, instantaneous frequencies f_k and phase offsets $\Phi_{k,0}$. A natural counterpart of additive synthesis is subtractive synthesis which carves the spectrum of a broadband excitation (e.g. white noise) at specific energy bands in order to generate a target spectral envelope. It can be performed from the spectrum domain with iSTFT (random phase) or it can be performed in the signal domain with a tuned bank of band-pass filters (classical vocoder) or by estimating the corresponding linear time invariant

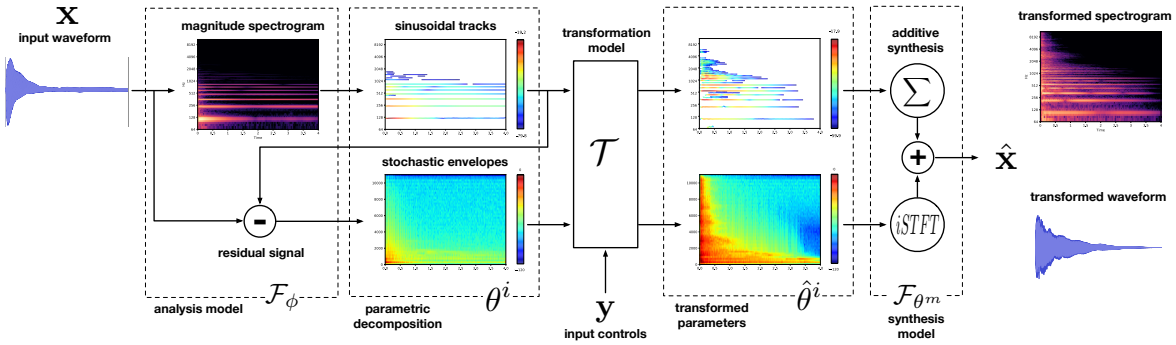

Figure 8: Analysis and synthesis (simplified diagram) in the context of the Sinusoidal plus Noise (SMS) decomposition. The input waveform is first converted to spectrogram and sinusoidal components are extracted (colouring by magnitude). The deterministic signal is subtracted from the input and the stochastic spectral envelopes are fitted on the residual signal. This yields the audio parameters of the SMS analysis stage, which can be transformed according to some user controls and inverted back to signal domain by additive synthesis plus inversion of the residual stochastic envelopes (as an alternative to subtractive synthesis). Analysis features are computed with the author's tools [236] `https://github.com/MTG/sms-tools`.

finite impulse response (LTI-FIR) filter to apply to the noise source [169]. Subtractive synthesis stems from the natural process of speech production in which the broadband glottal excitation is adaptively filtered by the human vocal tract and mouth in order to generate different phonemes. From the musical viewpoint, specific source-filter models have been proposed in order to allow independent control over pitch, loudness (the source) and timbre (the filter distribution) [28]. Additive and subtractive sound models are combined in the Spectral Modelling Synthesis (SMS [236]) technique which is an analysis/synthesis framework that accounts for both deterministic components such as harmonic partials and stochastic components such as transients and percussive features. The SMS model postulates that deterministic signals are slowly-varying narrow band energy components in the spectrogram (the analysis space from which synthesis parameters are extracted) that can be modelled with additive synthesis. For that purpose, the algorithm performs peak picking at each frame (extracting the frequencies, amplitudes and phases of)sinusoidal components) and peak tracking across frames so that variations of each deterministic signal are continuously modelled with the corresponding oscillator. The sum of deterministic signals is subtracted to the input audio and the residual stochastic signal is fitted with a spectral envelope that is used for subtractive synthesis.

The extracted parameters can be modified prior to synthesis in order to indepen-

dently alter the pitch (e.g. shifting the frequencies) or to perform timbre transformations such as sound morphing in between synthesis parameters extracted from two distinct sources [285]. There exists a trade-off as an accurate fitting of the quasi-sinusoidal signals requires a large analysis window (increased frequency resolution) whereas fine modelling of transients requires a short analysis window (increased time resolution). A precise SMS decomposition can be performed with separate analysis windows for fitting these two components (either smooth time or smooth spectrum), although the final representation complexity increases in terms of dimensionality and number of parameters. Further extensions have been proposed based on the SMS model, notably using a dedicated transient model [276] [92] or jointly modelling stationarity and stochasticity in the full-range spectrum with modulated oscillators [29] [78]. On the other hand further factorisations can be made such as assuming harmonic relationships between partials, yielding a more compact/lossy representation which is more practical to visualise and process. Generally speaking, the accuracy and expressivity of spectral models (SMS, vocoders) is related to their complexity and the generalisation power bound to that of the chosen feature decomposition (e.g. ability to reconstruct a specific subset of sounds or broader corpora).

Since there exists a huge gap in temporal dimensionality from compressed abstract music representations and the corresponding waveform rendering, audio synthesis may be performed in a hierarchical fashion such as acoustic production in the intermediate domain of spectrogram coefficients (conditioned on high-level musical context) $|\mathbf{X}| = \mathcal{F}_{\theta^{m_1}}(\mathbf{y})$ and then audio synthesis from spectro-temporal features $\mathbf{x} = \mathcal{F}_{\theta^{m_2}}(|\mathbf{X}|)$ (e.g. inversion with GLA). The magnitude spectrogram has a 2D acoustic structure which resembles that of the score (e.g. visualised as a piano-roll) as well as a regular feature distribution over time. Given the frame-wise segmentation and alignment of the musical context, the acoustic features of the spectrogram can be up-sampled to waveform by a dedicated model which is responsible for the local signal properties. It is thus a convenient intermediate representation in the music generation process and a base for audio synthesis parameter extraction. Yet it remains an open challenge to match existing purely analysis techniques (e.g. semantic extraction of music properties, perceptual timbre spaces) with synthesis in order to condition generation on high-level contexts and limitations imposed by classical analysis/synthesis models often prevent processing complex recordings (e.g. music mixture, complete performance).

2.4 Machine Learning for Music and Audio Processing

Throughout this section, music and audio processing have been introduced through a hierarchy of representations and handcrafted analysis/synthesis stages that either extract and classify higher-level music properties (retrieval) or render the acoustic information corresponding to such musical contexts and user controls (generation). The engineering of a technique may be broadly divided in two steps, the first is feature design and selection in order to emphasise the relevant data properties, the second is task modelling using some theoretical knowledge and intuition on how to process the chosen data representation (e.g. classification rules). The definition and refinement of a technique often requires many trial and error until sufficient fine-tuning of the configuration. Novel data-centred approaches have been first applied in the field of Music Information Retrieval (MIR) using Machine Learning (ML) models to replace task-specific engineered systems [123]. ML architectures make use of generic non-linear functions whose parameters can be trained for a given set of observations and corresponding task objective. Given a sufficient capacity (number of internal parameters), ML models are universal approximators [115] which hold the promise of outperforming classical audio engineering. Modern ML models are iteratively optimised by gradient descent of a loss function (objective) which assesses the performance of the current model state in order to update its parameters towards the loss decrease. For supervised predictive tasks such as MIR, we consider a dataset $\mathcal{X} = \{\mathbf{x}_i\}$ of observations (e.g. audio snippets) with corresponding ground-truth labels $\mathcal{Y} = \{\mathbf{y}_i\}$ (e.g. musical properties). The parametric model $\mathcal{F}_\phi$ is tasked to infer an annotation $\hat{\mathbf{y}} = \mathcal{F}_\phi(\mathbf{x})$ and a loss function $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$ measures the prediction error with respect to the supervised target $\mathbf{y}$. The model state at a given optimisation step is defined by its parameters ϕ which are updated by gradient descent of the differentiable loss function such that $\hat{\phi} = \phi - \eta \nabla_\phi [\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})]$ and $\phi \leftarrow \hat{\phi}$ until convergence, with η an hyper-parameter to adjust the learning rate. Deep learning models [94] are built by stacking multiple layers which are composed as $\mathcal{F}_\phi = f_{\phi_D} \circ \ldots \circ f_{\phi_1}$ with a depth D (number of layers) and optimisation is performed by back-propagation of the loss gradient from the last layers (output) to the firsts (input) (Figure 9). Stochastic updates of the model state are iteratively performed by drawing random mini-batches of data samples and annotations at each training iteration. Each layer f_{ϕ_d} with $d \in [1, D]$ is usually defined as the composition of a linear transformation (e.g. weight matrix $\mathbf{W}$ and bias vector $\mathbf{B}$) with a non-linear activation σ and the layer stack goes from input $\mathbf{x}$ to output $\mathbf{y}$ by com-

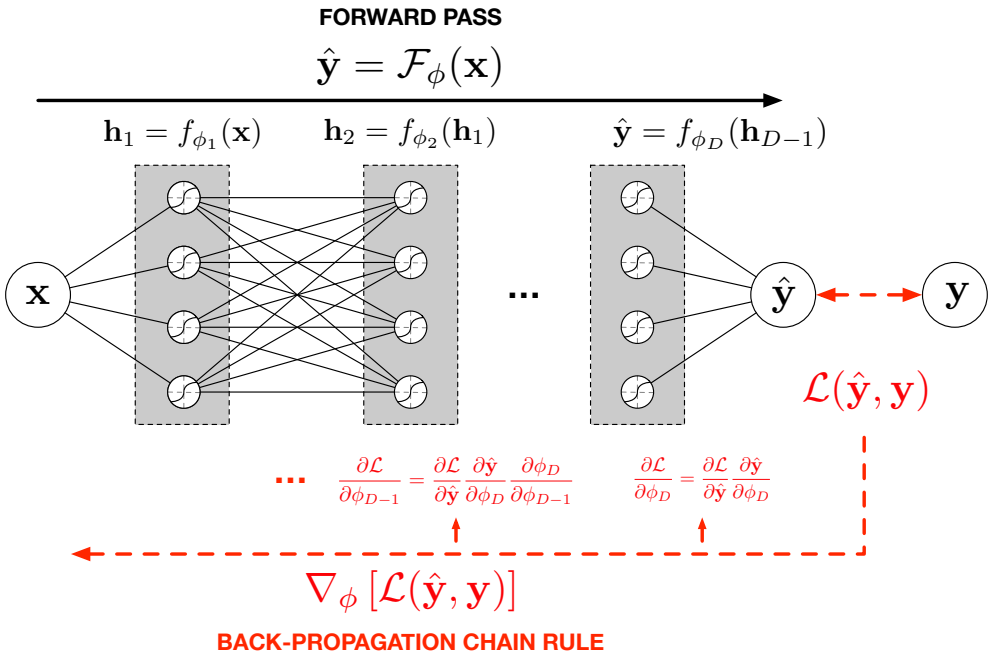

Figure 9: Forward pass and error back-propagation for training a generic supervised deep learning model by gradient descent. Stochastic optimisation is iteratively performed on random mini-batches of data until convergence of the model.

puting hidden features as $\mathbf{h}_{d\in[1,D-1]} = \sigma_d(\mathbf{h}_{d-1}^T \mathbf{W}_d + \mathbf{B}_d)$. Deep learning architectures are thus able to extract data features through a cascade of learnable transformations, from low-level features in the first layers to more abstract features that are aggregated into the output prediction of a given high-level target (e.g. label classification).

Supervised music information retrieval tasks are mainly based on audio analysis and do not require a subsequent synthesis. This analysis proceeds by down-sampling and extraction of features that are ultimately classified either as global labels (e.g. music tagging) or as predictions over time frames (e.g. transcription, segmentation). Since the inversion to audio is not required, it is common to pre-process the data into short-term features such as magnitude spectrograms (e.g. CQT) which offer a more structured input representation than the raw waveform for neural network training and pattern recognition. Convolutional neural networks and pooling have been widely used on magnitude spectrograms [41] to extract translation invariant features (e.g. onset detection) that are fed into classification layers such as fully connected and recurrent neural networks that model global relationships and summarise these features into the output predictions (e.g. tempo estimation). An overview of common tasks and

architectures for MIR is available in the following tutorial [40] along with reference datasets [1] and evaluation benchmarks [2].

Besides MIR, some signal-domain audio processing tasks have been increasingly tackled with deep learning [212] amongst which are source separation, extracting individual tracks from an audio mixture, and audio enhancement (e.g. denoising). These tasks may be categorized as transformations because their input and output representations are usually the same. Since they ultimately target the audio signal as an output, dedicated neural network architectures have been developed to carry the processing on the raw waveform [251] without need of lossy inversion of an intermediate output prediction such as magnitude spectrogram. In the recent MIR advances we also observe novel approaches that incorporate signal-domain processing to improve discriminative tasks, for instance as a learned pre-processing for automatic music transcription in [202] or as a mean to tackle unsupervised learning [38] on unannotated datasets with losses based on audio reconstruction.

Amongst audio and music processing tasks, generation with deep learning models is a growing field of research [24] which raises challenges of evaluation [294] (e.g. creative and perceptual quality) as well as development of adapted architectures. Generative modelling of music requires learning long temporal scales and complex semantic structures (e.g. polyphony, mixtures) that were restricted to symbolic representations and music composition with neural networks adapted from natural language processing [118]. These approaches to automatic music generation could be successfully extended to the waveform domain [53] using extremely large neural network models to synthesize whole songs imitating popular music genres. Although it is a significant technological feat, we may observe several limitations that generally apply to the approaches of automatic music generation, from which we distance in this thesis research. These models are often computationally intensive and little interpretable, whereas the music generation task may be divided into more efficient and modular networks performing sub-tasks such as melody and chord generation, accompaniment and drum generation, singing voice synthesis, individual instrument synthesis. Moreover, such systems offer little to no control which hinders creative applications and subjective values driven by user interactions.

[1] `http://ismir.net/resources/datasets/`

[2] `https://www.music-ir.org/mirex/wiki/MIREX_HOME`

3 Deep Generative Modelling Frameworks

3.1 Problem formulation

3.1.1 Unsupervised learning

In the previous section 2.4, the use of machine learning has been introduced in the framework of information retrieval. Data-driven MIR tasks are usually tackled with annotated databases which pair data observations with target properties in order to provide supervision to the training of a classification or regression model. In this section we introduce unsupervised learning with deep generative models (DGMs), a framework which does not rely on learning a mapping between human-annotated input and output pairs. Instead DGMs aim at learning factors of variations from the data observation themselves by probabilistic modelling. Unsupervised learning benefits from the large availability of unlabelled data, moreover it frees the model from reproducing potential biases of human assessment and allows learning generic representations of the data which may be transferred to multiple down-stream tasks. A given set of N data observations $\mathcal{X} = \{\mathbf{x}_{i=1\dots N}\}$ is assumed to be drawn independently from an underlying distribution $p(\mathbf{x})$ which the DGM aims at estimating in order to generate novel yet consistent data samples. Thus, the DGM parametrises a family of distributions $p_\theta(\mathbf{x})$ to be fit on the observations so that $p_\theta(\mathbf{x}) \approx p(\mathbf{x})$ and sampling of new data could be performed by $\hat{\mathbf{x}} \sim p_\theta(\mathbf{x})$.

The frequentist approach. A common estimator in the parameter space of θ is the Maximum Likelihood Estimation (MLE), i.e. maximising the probability of the observed data under the independent identically distributed (iid) assumption

$$\theta^* = \arg\max_\theta p_\theta(\mathcal{X}) = \arg\max_\theta \prod_{i=1}^{N} p_\theta(\mathbf{x}_i). \tag{8}$$

The optimum parameter configuration is found at the global maximum of the likelihood function $p_\theta(\mathcal{X})$ such that $\frac{\partial p_\theta(\mathcal{X})}{\partial \theta} = 0$. In practice a more convenient objective is the log-likelihood, given that log is a monotonically increasing function any positive-valued

function $p_\theta(\mathcal{X})$ (e.g. probability function) satisfies

$$\begin{aligned} \arg\max_\theta p_\theta(\mathcal{X}) &= \arg\max_\theta \ \log\left(p_\theta(\mathcal{X})\right) \\ \theta^* = \arg\max_\theta \sum_{i=1}^{N} \log\left(p_\theta(\mathbf{x}_i)\right) &= \arg\max_\theta \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \log\left(p_\theta(\mathbf{x})\right). \end{aligned} \tag{9}$$

The log-likelihood conveniently turns the product of many probabilities into a sum which eases numerical computation and can yield a concave optimisation space that ensures a unique solution $\frac{\partial \log(p_\theta(\mathcal{X}))}{\partial \theta} = 0$ which is the global maximum. Since we do not have access to the true data distribution, MLE matches the model distribution to the empirical data distribution defined by the training set which is found to be asymptotically convergent as the number of data observations tend to infinite $N \to \infty$, given that the unknown target distribution $p(.)$ lies within the model family $p_\theta(.)$. This yields a robust parameter estimator, however its expressiveness is limited to the families of parametric distributions which have a closed-form expression and most real-world datasets are far too complex to lie within tractable models. Amongst usual families of parametric probability distributions (see [94] Chapter 3.9) are:

name	**variables**	**expression**
Bernoulli	$x \in \{0, 1\}$ (binary) $0 < \theta < 1$	$p(x = 1) = \theta$ $p(x = 0) = 1 - \theta$ $p(x; \theta) = \theta^x (1 - \theta)^{(1-x)}$
Categorical	$x \in \{1, K\}$ (K discrete states) $x_i = 1$ when $x = i$ $0 < \theta_i < 1$	$p(x = i) = \theta_i$ $\sum_{i=1}^{K} \theta_i = 1$ $p(x; \theta) = \prod_{i=1}^{K} \theta_i^{x_i}$
Gaussian	$x \in \mathbb{R}$ $\theta = \{\mu, \sigma\}$ $\mu \in \mathbb{R}$ and $\sigma > 0$	$\mathcal{N}(x; \mu, \sigma) =$ $\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}\right)$

Common parametric distributions for univariate random variables.

The Bayesian approach. The density estimation by MLE yields a single parameter configuration θ^* as a function of the dataset which is considered as random observations sampled from an underlying data probability. Another approach based on Bayesian statistics treats the parameters θ as random variables in order to account for the uncertainty of knowledge given the finite dataset of observations. In this setting, the model is

defined by the joint probability $p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$ where the prior $p(\theta)$ is chosen with a large entropy that reflects the uncertainty over θ before observing any data. On the other hand, the posterior $p(\theta|\mathbf{x})$ reflects the knowledge over parameters gained by data observation (decrease in entropy) and can be derived from the conditional likelihood $p(\mathbf{x}|\theta)$ according to the Bayes' rule:

$$\begin{aligned} p(\theta|\mathbf{x}) &= \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \\ \underbrace{p(\theta|\mathcal{X})}_{\substack{\text{aggregated} \\ \text{posterior}}} &\propto \underbrace{p(\mathcal{X}|\theta)}_{\substack{\text{dataset} \\ \text{likelihood}}} \underbrace{p(\theta)}_{\substack{\text{model} \\ \text{prior}}} \end{aligned} \tag{10}$$

where $p(\mathbf{x}) = \int p(\mathbf{x}, \theta)\ d\theta$ refers to the marginal likelihood. In this setting, parameter estimation is posed as a posterior distribution which is given by the data observations and a prior which traduces an initial modelling choice (e.g. human assumptions or computational convenience). Recalling the goal of sampling new data $\hat{\mathbf{x}}$, the Bayesian approach allows two predictions using either the model posterior or prior distributions:

$$\begin{aligned} p_{post}(\hat{\mathbf{x}}|\mathcal{X}) &= \int p(\hat{\mathbf{x}}|\theta)p(\theta|\mathcal{X})\ d\theta \\ p_{prior}(\hat{\mathbf{x}}) &= \int p(\hat{\mathbf{x}}|\theta)p(\theta)\ d\theta. \end{aligned} \tag{11}$$

However given that the fitted posterior has a reduced entropy, it can be desirable to have a point estimate of the parameters rather than a probability over which to integrate in order to make predictions. The common choice of parameter estimate is then given by the maximum a posteriori:

$$\theta^* = \arg\max_{\theta}\ \log\left(p(\theta|\mathcal{X})\right) = \arg\max_{\theta}\ \log\left(p(\mathcal{X}|\theta)\right) + \log\left(p(\theta)\right). \tag{12}$$

Different learning frameworks have been proposed to model real-world datasets with a sufficient expressivity, at the expense of tractability. Some of these, which will be detailed in the next section, involve likelihood-based objectives and Bayesian inference whereas others perform implicit density estimation. Unsupervised learning in the audio domain notably requires modelling temporal dependencies within time-series and scaling to highly dimensional datasets. This traduces in modelling the density $p(\mathbf{x}) = p(x_0, \ldots, x_{L-1})$ spanning a large number L of waveform samples (e.g. hundreds

of thousands) for each observation in an audio library of arbitrary size N.

3.1.2 Conditional density estimation

Many applications of DGMs involve a certain context from which data samples depend on, notably some higher-level controls over the expected data properties being generated. This process corresponds to conditional density estimation, which introduces some sort of supervision as the model is given a dataset $\mathcal{X}$ which should relate to certain input features $\mathcal{Y}$ in order to learn to generate new consistent data under the constraints of a given combination of conditioning parameters. Considering an input variable $\mathbf{y}$ (e.g. a certain class of data observations), conditional density estimation aims at modelling the subsequent probability of sampling $\hat{\mathbf{x}}_{\mathbf{y}} \sim p(\mathbf{x}|\mathbf{y})$. Considering $\mathbf{y}$ as a random variable, this conditional probability can be computed as:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} \tag{13}$$

Assuming that the input distribution does not depend on the subsequent generative process, a maximum likelihood estimate is then defined as:

$$\theta^* = \arg\max_{\theta} p_\theta(\mathcal{X}|\mathcal{Y}) = \arg\max_{\theta} p_\theta(\mathcal{X}, \mathcal{Y}). \tag{14}$$

To the extent of neural audio synthesis, we may consider some higher-level contexts such as score, instrumentation, genre and a degree of performance variability under which exists a certain variability in terms of potential audio outputs. The desired properties of the conditional generative model are its realisticness, the synthetic data should be indiscernible from real data, accuracy, it should give outputs which are perceived as belonging to the target class and range of the data, and the diversity of its outputs given the conditioning context. These qualities parallel those of the evaluation of classical audio synthesis and its controllability [130]. In both cases, we may observe some trade-off such as that of accuracy and diversity.

3.2 Frameworks

In this section, we describe the main frameworks (Figure 10) for deep generative modelling with directed probabilistic models. These families of unsupervised models share the common goal of estimating a parametric density underlying a dataset of observations and can be extended to learning semi-supervised conditional generation. Several methods have been developed to tackle the task of unsupervised generative modelling, which mostly differ according to:

- ▷ which random variables are considered and how is their joint distribution decomposed
- ▷ how the model is parametrised and which training objective is used to estimate these parameters
- ▷ is the model learning some latent representation of the data and an inference mechanism (e.g. analysis and synthesis)

In this review of frameworks, we adopt the following notation:

$$\mathbf{x} \equiv \text{stochastic observed variable}$$
$$p_{\mathcal{X}} \equiv \text{empirical distribution of the dataset}$$
$$\mathbf{z} \equiv \text{stochastic unobserved/latent variable}$$
$$\theta \equiv \text{generator/decoder parameters e.g. } G_\theta \ ; \ p_\theta(\mathbf{x}|\mathbf{z})$$
$$\phi \equiv \text{inference/encoder parameters e.g. } E_\phi \ ; \ q_\phi(\mathbf{z}|\mathbf{x})$$
$$\psi \equiv \text{discriminator parameters e.g. } D_\psi \ ; \ p_\psi(y|\mathbf{x}).$$
$$y \equiv \text{discriminating and conditioning variables e.g. labels}$$

3.2.1 Fully-observed models

A fully-observed probabilistic model is such that all stochastic variables are being observed, to the extent of an unsupervised generative model this amounts to (only) the data observations. Throughout this section, we set the notations such that data observations $\mathbf{x} = \{x_0, \ldots, x_{L-1}\} \in \mathbb{R}^L$ belong to a dataset $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Accordingly,

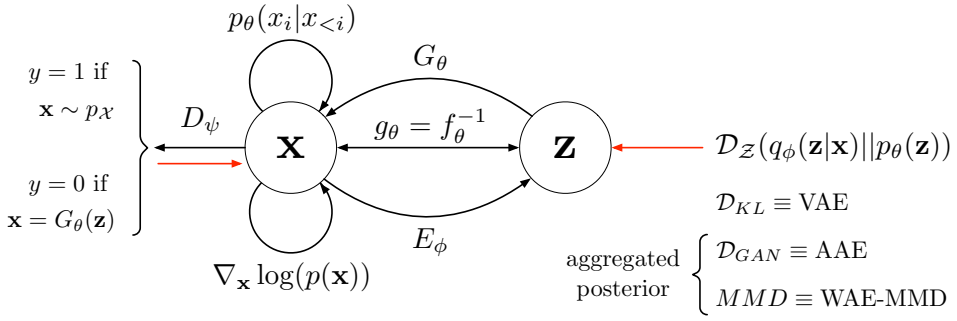

Figure 10: Overview of the main deep learning frameworks for unsupervised generative modelling. Fully-observed models only consider the probabilistic process of the data observations (loops over $\mathbf{x}$), which can be decomposed as an auto-regressive conditional density or estimated by its gradient field (score-matching). A joint distribution of the data and a stochastic representation $\mathbf{z}$ can be learned with latent variable models which condition the generative process G_θ based on a prior $p_\theta(\mathbf{z})$. The Generative Adversarial Networks (GAN) learn this model with an auxiliary discriminator D_ψ which classifies whether samples are ground-truth observations from the dataset or synthesised by the generator. On the other hand, approximate posterior inference can be learned with an encoder E_ϕ and a latent regularizer $\mathcal{D}_\mathcal{Z}$. It includes the Kullback-Leibler divergence $\mathcal{D}_{KL}$ used in the Variational Auto-Encoder (VAE) and aggregated posterior regularizers such as the Maximum Mean Discrepancy (WAE-MMD) or an adversarially learned divergence $\mathcal{D}_{GAN}$ (AAE). A tractable latent variable model can be learned with exact likelihood estimation using invertible neural networks $g_\theta = f_\theta^{-1}$ that can be stacked into Normalizing Flows. These base frameworks can be combined in order to take advantage from their respective strengths, for instance Variational Inference with a posterior parametrised with a normalizing flow, Adversarially Learned Inference with a joint discriminator over both data and latent distributions.

the multivariate data density of interest is $p(\mathbf{x}) = p(x_0, \dots, x_{L-1})$ and it can be decomposed into a product according to the chain rule:

$$p(x_0, \dots, x_{L-1}) = \prod_{i=1}^{L-1} p(x_i|x_{i-1}, \dots, x_0)p(x_0) = \prod_{i=1}^{L-1} p(x_i|x_{<i})p(x_0). \tag{15}$$

This equation is the basis of auto-regressive (AR) models that estimate the probability of each data dimension conditioned on the previously observed ones. This notably applies to time-series with causal dependencies because each time step is only conditioned on the past and the chain rule (15) preserves such ordering. The AR model task is thus to learn the conditional dependence which is usually parametrised with a neural network G_θ such that $p_\theta(x_i|x_{<i}) = p(x_i|G_\theta(x_{<i}))$, usually this network is shared across time steps thus we do not consider indexing θ_i. The different AR models mainly differ on how the past context is aggregated in order to make every next step predictions.

Auto-regressive modelling with a temporal auto-encoder. The Neural Auto-regressive Distribution Estimator (NADE [266]) uses a two-layer neural network to compute the conditional dependence based on the single preceding time step and a deterministic hidden state $\mathbf{h}$ such that for every $i \in [1, L-1]$:

$$\begin{aligned} \mathbf{h}_i &= G_{\theta^0}(x_{i-1}, \mathbf{h}_{i-1}) \\ \hat{x}_i &= G_{\theta^1}(\mathbf{h}_i). \end{aligned} \tag{16}$$

The model is trained by maximum likelihood, or equivalently by minimising the negative log-likelihood such that:

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{n=1}^{N} -\log\left(p_\theta(\mathbf{x}_n)\right) = \arg\min_\theta \frac{1}{N} \sum_{n=1}^{N} \sum_{i=0}^{L-1} -\log\left(p_\theta(x_i^{(n)} | x_{<i}^{(n)})\right). \tag{17}$$

A common challenge for AR models is to account for long-term dependencies by allowing the past to influence the current time step prediction at several orders of time steps back. This range called receptive field is usually fixed for architectural reasons (as neural networks process inputs of fixed dimensions) and its size T is limited by the finite computation resources allowed such that $p_\theta(x_i|x_{<i}) = p(x_i|G_\theta(x_{i-1}, \ldots, x_{i-T}))$. Thus AR models are often designed in order to maximise the receptive field for a given model complexity (i.e. number of parameters). One common approach to maximise the receptive field is to use stacked CNNs (weight sharing over time) with stride for down-sampling in between layers and enabling the upper convolutions to span a large context (e.g. ConvNADE).

Auto-regressive modelling with recurrent neural networks. Another approach to AR modelling is to use RNNs which have an internal memory that implicitly stores all the preceding information. Although RNNs could theoretically allow for any receptive field, practical applications face several challenges such as vanishing gradients which gradually extinguish the memory of past events. Several RNN cells have been proposed in order to alleviate this issue, amongst which the Long short-term Memory (LSTM [114]) is a common and efficient choice. The LSTM is comprised of a cell state $\mathbf{c}$ that stores information about the past and several gates that regulate the information flow from input to hidden state $\mathbf{h}$. Similarly to NADE (Figure 11), the hidden state is often used for the subsequent predictions such that $p_\theta(x_i|x_{<i}) = p_\theta(x_i|\mathbf{h}_{i-1})$. The LSTM

updates its hidden state according to:

$$\begin{aligned}
\text{input gate: } \mathbf{i}_i &= f_{input}(x_i, \mathbf{h}_{i-1}) \\
\text{forget gate: } \mathbf{f}_i &= f_{forget}(x_i, \mathbf{h}_{i-1}) \\
\text{cell gate: } \mathbf{g}_i &= f_{cell}(x_i, \mathbf{h}_{i-1}) \\
\text{output gate: } \mathbf{o}_i &= f_{output}(x_i, \mathbf{h}_{i-1}) \\
\text{cell state update: } \mathbf{c}_i &= \mathbf{f}_i \odot \mathbf{c}_{i-1} + \mathbf{i}_i \odot \mathbf{g}_i \\
\text{hidden state update: } \mathbf{h}_i &= \mathbf{o}_i \odot \tanh(\mathbf{c}_i)
\end{aligned} \tag{18}$$

with $\odot$ denoting the Hadamard product.

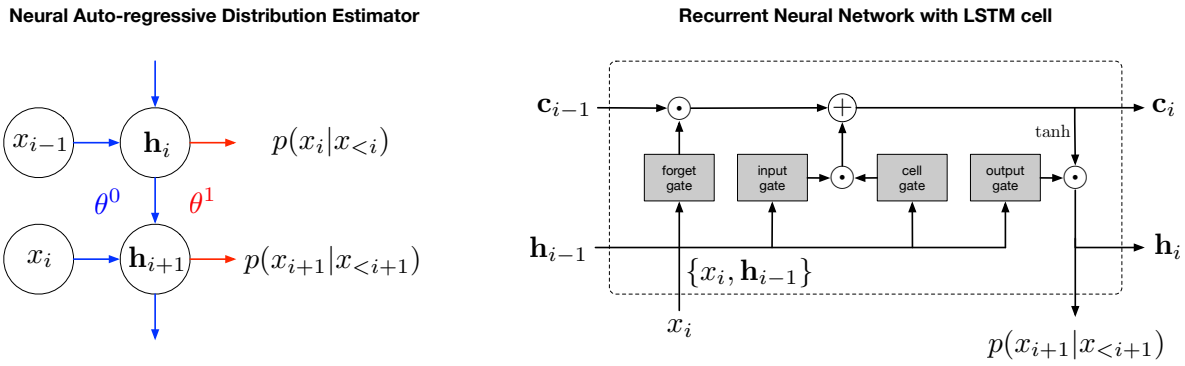

Figure 11: Unsupervised density estimation with the Neural Auto-regressive Distribution Estimator (left) and a LSTM Recurrent Neural Network (right).

Deterministic auto-encoders. In the case of NADE, the hidden activation $G_{\theta^0} : (x_{i-1}, \mathbf{h}_{i-1}) \rightarrow \mathbf{h}_i$ allows temporal dependency in the prediction of the next auto-regressive step $G_{\theta^1} : \mathbf{h}_i \rightarrow x_i$. Other types of auto-encoders make use of a deterministic hidden code for various analysis and synthesis purposes. In this setting a standard (non auto-regressive) auto-encoder is commonly defined as a pair of networks that learn to invert each other, the encoder $E_\phi : \mathbf{x} \rightarrow \mathbf{h}$ (analysis) and the decoder $G_\theta : \mathbf{h} \rightarrow \mathbf{x}$ (synthesis) which jointly optimise some reconstruction error $\mathcal{L}$ such that:

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} \mathcal{L}\left(\mathbf{x}, G_\theta(E_\phi(\mathbf{x}))\right) \tag{19}$$

In this regard, it should be noted that the unsupervised reconstruction objective does not always rely on a probabilistic decoding and an objective such as the log-likelihood (e.g. deterministic decoder trained with MSE). Regardless of the nature of its recon-

struction objective, the auto-encoder is usually designed so that its hidden code enforces certain learning properties [264], for instance:

- ▷ Dimensionality reduction [113], with a compression such that $\mathbf{h} \in \mathbb{R}^{d_h}$ and $\mathbf{x} \in \mathbb{R}^{d_x}$ with $d_h \ll d_x$.
- ▷ Denoising [277], with an invariance to small random input perturbations ϵ such that $G_\theta\left(E_\phi(\mathbf{x}+\epsilon)\right) \approx \mathbf{x}$. This invariance property can as well be learned by the contractive auto-encoder [222] which adds a penalty term to the loss function $\mathcal{L}\left(\mathbf{x};\theta,\phi\right) + \lambda||\nabla_{\mathbf{x}}\mathbf{h}||$ where λ is an hyper-parameter weighting how strongly the hidden code should be insensitive to the data.
- ▷ Sparsity [8], forcing that a reduced number of hidden dimensions h_i are active at once. This can be done by adding a $L1$ penalty to the loss function $\mathcal{L}\left(\mathbf{x};\theta,\phi\right) + \lambda\sum_{i=1}^{d_h}|h_i|$ which constrains the hidden code magnitudes.
- ▷ Clustering [247], which aims at grouping similar data observations into neighbouring hidden codes. This can be done using a mixture of auto-encoders (one per cluster) [298] or a supervised contrastive loss [136] given that data labels are available.
- ▷ Orthogonality of the hidden code dimensions [279] [155], akin to a Principal Component Analysis (PCA [83]), which can be achieved by adding a covariance penalty to the loss over data batches $\mathbf{X} \in \mathbb{R}^{M*d_x}$ and $\mathbf{H} \in \mathbb{R}^{M*d_h}$ so that $\mathcal{L}\left(\mathbf{X};\theta,\phi\right) + \lambda||\mathbf{H}^T\mathbf{H} - \mathbb{I}||$ where $\mathbb{I}$ is the identity matrix.

Score-based generative modelling. An alternative to maximum likelihood estimation is proposed by learning the gradients of the data distribution rather than the distribution itself, which is termed as the score $\nabla_{\mathbf{x}}\log(p(\mathbf{x}))$. The score matching [124] objective of a model G_θ is thus to minimise $\mathbb{E}_{\mathcal{X}}||G_\theta(\mathbf{x}) - \nabla_{\mathbf{x}}\log(p(\mathbf{x}))||_2^2$ which would lead to an iterative data generation process by ascending the direction of maximum score as estimated by the model (i.e. maximising the likelihood of a data sample initialised randomly). Since the actual data distribution cannot be estimated in closed-form, the naive definition of the score matching objective is neither tractable. Under some common regularity conditions, the score matching objective can be re-written [250] as:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\mathcal{X}}\ tr(\nabla_{\mathbf{x}} G_\theta(\mathbf{x})) + \frac{1}{2}||G_\theta(\mathbf{x})||_2^2 \tag{20}$$

with $tr()$ denoting the trace operator. In this setting, the model implicitly learns the vector field of the gradients of the data probability $G_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log(p(\mathbf{x}))$ while remaining fully tractable. The training does not require sampling, yet the model can be used in a generative setting by random initialisation $\mathbf{x}_0 \sim \mathcal{N}(\mathbb{O}, \mathbb{I})$ and implicit gradient ascent as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta\ G_\theta(\mathbf{x}_t). \tag{21}$$

The first term of equation 20 does not scale to deep networks nor high data dimensionality because of the computation cost of the Jacobian. Ongoing research shows promising advances for efficient score-based generative modelling [250] [249] [248] which compete with state-of-art results such as auto-regressive density estimation.

3.2.2 Latent variable models

In the fully-observed setting, the stochastic variables being modelled are only those of data observations which are approximated by $p_\theta(\mathbf{x})$. For instance a hidden code $\mathbf{h}$ is introduced in NADE so that the model learns the temporal dependency between time-steps, yet it does not obey to any probabilistic model besides that of maximising the data likelihood. Some hidden (also named latent or unobserved) stochastic variables $\mathbf{z} \in \mathcal{Z}$ can be introduced for enhancing the model which is then estimating a joint density $p_\theta(\mathbf{x}, \mathbf{z})$. This Latent Variable Model (LVM) can leverage some prior assumptions $p_\theta(\mathbf{z})$ on the data structure such that it learns a conditional density as $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$. As an example, we can formulate a Gaussian Mixture Model (GMM) with K components in which each of the $k \in \{1, K\}$ class probability is given by $p_\theta(z = k) = \pi_k$. Under this assumption of a categorical latent variable z, with K classes as multivariate Gaussian distributions parametrised by μ_k, Σ_k, the LVM is given by:

$$\begin{gathered} p_\theta(\mathbf{x}|z = k) = \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k) \\ p_\theta(\mathbf{x}) = \sum_{k=1}^{K} p_\theta(\mathbf{x}|z = k)p_\theta(z = k) = \sum_{k=1}^{K} \pi_k\ \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k). \end{gathered} \tag{22}$$

In the general case of a continuous latent variable $\mathbf{z} \in \mathbb{R}^{d_z}$, the calculation of the marginal likelihood requires integrating $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})\ d\mathbf{z}$ and the LVM opti-

misation by maximum log-likelihood would be computed as:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \log\left(p_\theta(\mathbf{x})\right) = \arg\max_\theta \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \log\left(\int p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})\, d\mathbf{z}\right). \quad (23)$$

In practice, this objective is no longer tractable (e.g. we cannot compute $\mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z})} p_\theta(\mathbf{x}|\mathbf{z})$) and requires specialised learning algorithms. On approach to approximate the LVM is the Expectation-Maximisation (EM [58]) algorithm which requires a tractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$. The EM algorithm iteratively updates the model parameters until convergence according to two steps:

$$\begin{gathered}\text{Expectation step: } \mathbf{z} \sim p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x},\mathbf{z})}{p_\theta(\mathbf{x})}\ \forall \mathbf{x} \in \mathcal{X} \\ \text{Maximisation step: } \hat{\theta} = \arg\max_\theta \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z}|\mathbf{x})} \log\left(p_\theta(\mathbf{x},\mathbf{z})\right) \\ \text{Update } \theta \leftarrow \hat{\theta} \text{ and repeat until convergence } \hat{\theta} \approx \theta^*.\end{gathered} \quad (24)$$

The Variational Auto-Encoder. Another approach to learn a LVM is Variational Inference (VI [81]) which turns posterior inference into an optimisation problem, thus it no longer requires a tractable expectation step. In the VI framework, an approximate posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ with free variational parameters ϕ which should be fit to the unknown posterior by minimising the Kullback-Leibler (KL) divergence:

$$\begin{gathered}\mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) = \int q_\phi(\mathbf{z}|\mathbf{x}) \log\left(\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})}\right) d\mathbf{z} \\ \mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log(q_\phi(\mathbf{z}|\mathbf{x})) - \log(p(\mathbf{z}|\mathbf{x})).\end{gathered} \quad (25)$$

By applying the Bayes' rule to the posterior, we have $p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}$ which allows to rewrite the objective as:

$$\begin{gathered}\mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) = \log(p(\mathbf{x})) + \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log(q_\phi(\mathbf{z}|\mathbf{x})) - \log(p(\mathbf{x}|\mathbf{z})) - \log(p(\mathbf{z})) \\ \log(p(\mathbf{x})) - \mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log(p(\mathbf{x}|\mathbf{z})) - \mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\end{gathered} \quad (26)$$

Since the KL divergence is positive, this provides a lower bound to the likelihood, known as the Evidence Lower Bound (ELBO):

$$\log(p(\mathbf{x})) \geq \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log(p(\mathbf{x}|\mathbf{z}))}_{\text{conditional likelihood}} - \underbrace{\mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))}_{\text{regularizer}} \quad (27)$$

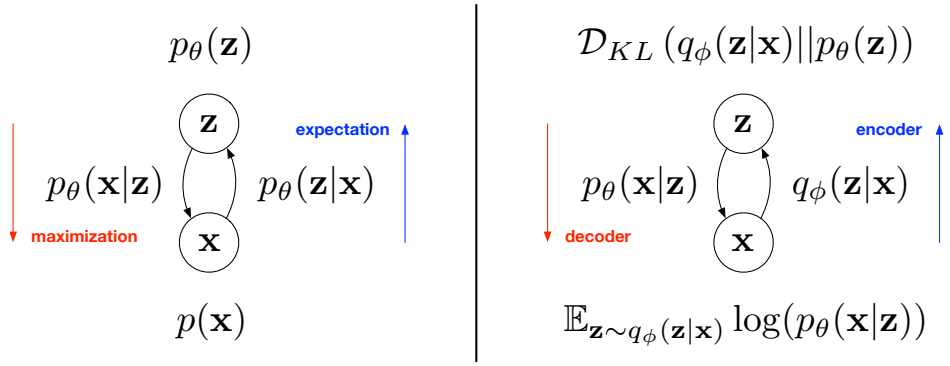

Figure 12: Learning a latent variable model by Expectation-Maximisation (EM, left) and Variational Inference (VI, right). The EM algorithm requires a tractable posterior distribution, the Variational Auto-Encoder implements VI by using an encoder network to approximate the posterior. End-to-end training can be performed by jointly optimising the expected conditional data likelihood and minimising the Kullback-Leibler divergence with the decoder prior.

which is comprised of a term that corresponds to maximising the conditional data likelihood and a regularizer that forces the variational posterior distribution to match the model prior. The VI framework has been implemented into the Variational Auto-Encoder (VAE [145]) by neural network parametrisation of both the variational posterior distribution with an encoder E_ϕ and the conditional generation with a decoder G_θ (Figure 12). In order to allow end-to-end learning of the model parameters θ, ϕ with stochastic back-propagation, the VAE should be parametrised such that we can analytically compute the KL divergence and the gradients through sampling from the latent posterior $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$. The vanilla VAE implements this with an isotropic unit variance Gaussian prior distribution $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbb{O}, \mathbb{I})$ and an inference posterior $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \sigma^2_\phi(\mathbf{x}))$ (diagonal covariance for mean field approximation) with parameters output by the encoder $E_\phi : \mathbf{x} \to \mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x})$ (amortised inference). Under these conditions, the sampling, KL divergence and training objective can be directly computed as:

$$
\begin{aligned}
\mathbf{z} &= \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \epsilon \text{ with } \epsilon \sim \mathcal{N}(\epsilon; \mathbb{O}, \mathbb{I}) \text{ (reparametrisation trick)} \\
\mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) &= 0.5 \sum_{i=1}^{d_z} \mu_i^2 + \sigma_i^2 - 1 - \ln \sigma_i^2 \\
\theta^*, \phi^* &= \arg\min_{\theta,\phi} \underbrace{-\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log(p_\theta(\mathbf{x}|\mathbf{z})) + \mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))}_{\mathcal{L}_{\text{ELBO}}(\mathbf{z}\sim E_\phi(\mathbf{x}), \hat{\mathbf{x}}\sim G_\theta(\mathbf{z}), \mathbf{x})}.
\end{aligned}
\tag{28}
$$

Representation learning and the manifold hypothesis. The original parametrisation of the VAE is justified by the ease of computation and differentiation of the forward pass through the auto-encoder network. Moreover, latent regularisation schemes give rise to representation learning properties [2] which can be interpreted in the context of the VAE and the manifold hypothesis [13]. To this extent, we consider the continuous space of audio waveforms $\mathbf{x} \in \mathbb{R}^L$ where L has a size in the order of macroscopic observations (e.g. 10^5 for a few seconds). Sampling the uniform distribution $\mathbf{x} \sim \mathcal{U}^L_{[-1,1]}$ will consistently produce white noise whereas all natural sounds are comprised within this same distribution. This justifies the hypothesis that natural data observations are embedded in a sub-space of lower dimensionality than the apparent data space and that the manifold dimensions are related to the underlying structure of real-world data (i.e. factors of variations) as opposed to the dimensions of the macroscopic observation space [77]. The VAE posterior regularisation with a simple latent prior distribution (e.g. isotropic Gaussian), often chosen with a reduced dimensionality such that $d_z \ll d_x$, forces the model to encode a smooth and compact representation of the data which relates to the manifold properties (Figure 13). This prior over the latent representation constrains inference and allows ancestral sampling $\hat{\mathbf{x}} \sim p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$ in which data generation in the observation space can be conditioned by a much simpler latent distribution.

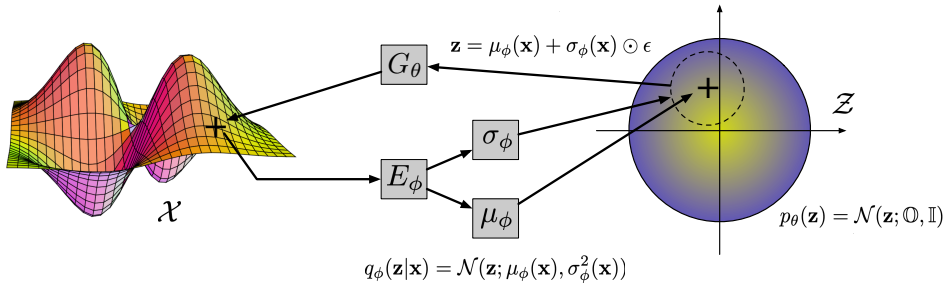

Figure 13: Manifold learning in the context of the Variational Auto-Encoder with mean field approximation. Natural data $\mathcal{X}$ are folded in a sub-space of the observation space which the encoder maps to the dense latent space $\mathcal{Z}$ regularised by an isotropic Gaussian prior distribution.

Regularizers for continuous latent spaces. The Variational Auto-Encoder is a powerful framework for unsupervised representation learning and generative modelling, which has motivated extensive investigations in order to enhance its capacities [146]

[284]. Amongst follow-up research works carried with VAEs, we discuss a few findings and modifications of the original model implementation. The effect of a β hyper-parameter weighting on the KL divergence was studied in [111] [27] and trades-off the conditional data-likelihood with an increased regularizer strength which favours learning a disentangled latent representation under the mean field approximation. The experiments show that $\beta > 1$ encourages the encoder to use separate latent dimensions for the factors of variations underlying the subsequent generative process. The application in the visual domains demonstrates that higher-level properties such as shapes and dimensions can be manipulated via distinct latent dimensions. Nonetheless, the regularisation scheme introduced in the VAE is prone to failure cases which may be observed from the derivation of the ELBO. In equation 26 we can see that the estimation of the data likelihood could be unbounded if the variational posterior was exactly matched to the true latent posterior distribution. However its restriction to the mean field family of isotropic Gaussian distributions is unlikely to comprise the true latent distribution of real-world datasets, causing a gap between the ELBO optima and the true data distribution. This limitation may be responsible for the fact that VAE samples tend to miss fine-grained features of the data distribution (e.g. blurry samples) [302]. Another issue, referred as over-pruning [296], is observed when the optimisation of the KL divergence pushes the posterior distribution to exactly match the prior. The latent dimensions which collapse to zero mean and unit variance minimise the KL divergence but become uninformative and over-pruning leads to a degenerate VAE in which the decoder by-passes the encoder.

The ELBO gap due to the mean field variational posterior and the failure case of over-pruning (posterior collapse) have motivated an alternative regularisation scheme in the Wasserstein Auto-Encoder (WAE [263], also introduced in the InfoVAE [303]). Considering a batch of data $\mathbf{X} \in \mathcal{X}$ and the corresponding batches of latent samples $\mathbf{Z}$, the WAE regularizer is applied over the aggregated latent distributions as $\mathbb{E}_{\mathcal{X}}\mathcal{D}_{\mathcal{Z}}(q_\phi(\mathbf{Z}|\mathbf{X})||p_\theta(\mathbf{Z}))$ where $\mathcal{D}_{\mathcal{Z}}$ can be an arbitrary divergence measure between the latent mixtures of posterior and prior samples. While the VAE regularisation pushes each individual latent code to match the prior, the WAE regularizer only forces the encoder posterior to match the prior in average. This enables a more flexible regularisation and the choice of an arbitrary divergence measure permits the use of a deterministic encoder as well as diverse prior distributions without requiring the analytical KL divergence calculation. In the WAE-MMD framework, the Maximum Mean Discrepancy

(MMD [97]) is used as the divergence measure $\mathcal{D}_{\mathcal{Z}}$ which is a type of kernel two-sample test. The MMD does not take assumptions on the parametric forms of the compared distributions, instead it uses their kernel estimates to evaluate their degree of similarity given batches of samples independently drawn from both. Considering two distribution P, Q and the corresponding batches of samples $\mathbf{X}, \mathbf{X}' \sim P$ and $\mathbf{Y}, \mathbf{Y}' \sim Q$ the MMD is computed as:

$$MMD(P,Q) = \underbrace{\mathbb{E}_P k(\mathbf{X}, \mathbf{X}')}_{\substack{\text{intra } P \\ \text{similarity}}} + \underbrace{\mathbb{E}_Q k(\mathbf{Y}, \mathbf{Y}')}_{\substack{\text{intra } Q \\ \text{similarity}}} - 2\underbrace{\mathbb{E}_{P,Q} k(\mathbf{X}, \mathbf{Y})}_{\substack{\text{cross } P,Q \\ \text{dissimilarity}}} \tag{29}$$

where the kernel k can be chosen as the radial basis function $k(x, x') = \exp -||x - x'||_2^2$. In the deterministic case, the WAE-MMD objective can thus be written as:

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} \mathbb{E}_{\mathcal{X}} c(\mathbf{X}, G_\theta(E_\phi(\mathbf{X}))) + MMD(E_\phi(\mathbf{X}), p_\theta(\mathbf{Z})) \tag{30}$$

with c any reconstruction cost (e.g. MSE).

Discrete representation learning. Another stream of research has focused on learning a LVM with a discrete latent prior that would potentially reflect the categorical nature of the data (e.g. a GMM model which assigns class probabilities to the latent variable). The Vector-Quantized Variational Auto-Encoder (VQ-VAE [270]) implements a discrete latent representation by defining the prior as a set of vectors $\mathbf{Q} = \{\mathbf{q}_{i=1\ldots K}\} \in \mathbb{R}^{K * d_z}$ which are dynamically learned in an unsupervised fashion. The deterministic VQ-VAE encoding $E_\phi : \mathbf{x} \to \mathbf{z}$ is quantised into the latent codebook $\mathbf{Q}$ by nearest neighbour lookup, which yields a categorical posterior distribution and subsequent decoding of the form:

$$\begin{aligned} q_\phi(z = k|\mathbf{x}) &= \begin{cases} 1 & \text{for } k = \arg\min_{i \in [1,K]} ||\mathbf{z} - \mathbf{q}_i||_2 \\ 0 & \text{otherwise} \end{cases} \\ \hat{\mathbf{x}} &= G_\theta(\mathbf{q}_k). \end{aligned} \tag{31}$$

Because the nearest neighbour quantisation step is not differentiable, end-to-end back-propagation of the VQ-VAE loss is performed via straight-through approximation [14] which copies the gradients back-propagated at the decoder input to the encoder output

and allows the following optimisation:

$$\mathcal{L}_{\text{VQ-VAE}} = c(\mathbf{x}, \hat{\mathbf{x}}) + \underbrace{||sg(\mathbf{z}) - \mathbf{q}_k||_2^2}_{\mathcal{L}_{\text{codebook}}} + \beta \underbrace{||\mathbf{z} - sg(\mathbf{q}_k)||_2^2}_{\mathcal{L}_{\text{commitment}}}$$
$$\underbrace{\nabla_{\mathbf{q}_k,\theta}\, \mathcal{L}_{\text{VQ-VAE}}}_{\text{decoder gradients}} \rightarrow \underbrace{\nabla_{\mathbf{z},\phi}\, \mathcal{L}_{\text{VQ-VAE}}}_{\text{encoder gradients}} \tag{32}$$

with $sg(.)$ denoting the stop gradient operator (identity at forward and zeroing of gradients at backward). The second term $\mathcal{L}_{\text{codebook}}$ updates the selected codebook elements towards the unquantised encoder outputs (which is held constant by stop gradient) and the third term $\mathcal{L}_{\text{commitment}}$ inversely pushes the encoder outputs towards the nearest neighbour quantisation code (while freezing the codebook by stop gradient). Overall, the VQ-VAE objective is such that the decoder parameters optimise the first loss (reconstruction cost), the encoder parameters optimise the first and last losses, the codebook optimises the first and second losses. The KL divergence can be omitted from this objective by postulating a uniform prior distribution over the codebook $p_\theta(z = k) = 1/K\ \forall k \in [1, K]$ which amounts to a constant KL divergence $\mathcal{D}_{KL}(q_\phi(z|\mathbf{x})||p_\theta(z)) = \log(K)$. Because the KL divergence is held constant throughout the training, the VQ-VAE alleviates the issue of posterior collapse which often happens in the continuous VAE setting when using a powerful decoder that may by-pass the encoder inference. As an example the VQ-VAE can be trained on speech waveform with a powerful auto-regressive decoder which does not impede the learning of a rich latent representation, experiments in [270] [44] show that the discrete latent space can unsupervisedly embed features that strongly correlate with speech phonemes. Follow-up experiments have introduced alternative implementations of a Vector-Quantized latent space in order to increase the training stability of the original VQ-VAE. The codebook update loss from 32 can be replaced by an exponentially moving average or it can be removed if keeping the codebook as a constant set of one-hot vectors in the Argmax Auto-Encoder (AMAE [54]). In the aforementioned VQ-VAEs, the categorical posterior inference may lead to a poor use of the codebook which is another form of collapse when the model effectively uses only a small fraction of the codebook elements. A Soft-VQ-VAE is proposed in [289] with a decoder that outputs a categorical distribution over the codebook and posterior inference is performed with a mixture model of its K components.

Sequential and hierarchical representation learning. As shown in Figure 13, the VAE latent space mirrors the data space via posterior inference over individual and independent data observations. In this setting, each data point is individually assigned some latent parameters $\{\mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x})\}$ with a feed-forward encoder. In the case of some sequential data $\mathbf{x}^{(1,\dots,T)} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T)}\}$ the learned representation should capture the dependencies in the data in a global embedding $\mathbf{z}^{(g)}$ as opposed to individually encoding each step $\mathbf{z}^{(t)}$ for $t \in [1, T]$. The VAE framework can be applied to time series by using RNNs in the encoder and the decoder, as introduced in the Variational Recurrent Auto-Encoder (VRAE [76]) which learns an embedding over MIDI songs (series of note symbols). The VRAE is also applied to natural language in [23] in order to learn a global embedding of sentences which can be sampled in order to generate new sentences with coherent series of words. The model is comprised of an RNN-based encoder which uses its last hidden state to compute the global latent parameters as:

$$\begin{aligned} \mathbf{h}^{(t+1)} &= RNN_\phi(\mathbf{x}^{(t+1)}, \mathbf{h}^{(t)}) \\ E_\phi : \mathbf{h}^T &\rightarrow \mu_{\mathbf{z}^{(g)}}, \sigma_{\mathbf{z}^{(g)}}. \end{aligned} \tag{33}$$

Sampling can be performed using the reparametrisation trick of equation 28 and the RNN decoder hidden state is initialised using that embedding before auto-regressive prediction of the output series (Figure 14). This results in a sequence to sequence model [255] with a variational information bottleneck $\mathbf{z}^{(g)} \in \mathbb{R}^{d_z}$ that models the joint probability of sequential data $\mathbf{x}^{(1,\dots,T)} \in \mathbb{R}^{d_x * T}$. Several variations of recurrent VAEs for time-series modelling have been proposed [91], which share the same training objective (ELBO) but mainly differ in the graph structure linking the observed variables and the latent variables for both posterior inference, sampling and sequential prediction.

The VRAE tackles the learning of a global latent representation of sequential data, as opposed to individually encoding local features of each time step. The distinction between local and global features highly depends on the data structure (e.g. time-series or static observations) and the model, which may explicitly learn hierarchical representations that span different contexts and degrees of abstraction (Figure 14). A hierarchical VRAE is proposed in [37] which uses a pyramidal architecture with multiple encoder-decoder pairs processing different temporal scopes. The upper latent space aggregates a larger context and is complementary to a lower latent space that models shorter-term dependencies in the data. Hierarchical learning is also proposed for static

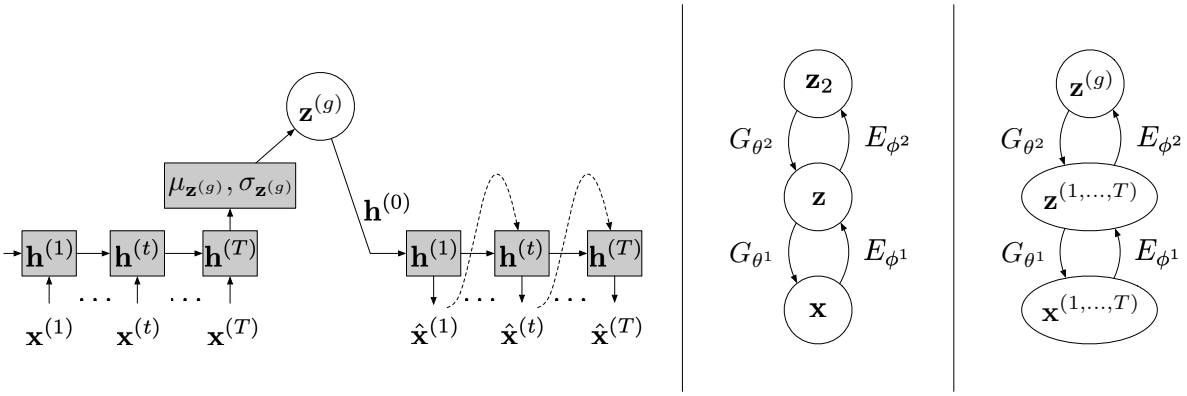

Figure 14: Left: A VRAE with a RNN encoder that summarises an input sequence into a final hidden state $\mathbf{h}^{(T)}$ that is used to infer a global latent $\mathbf{z}^{(g)}$. The RNN decoder is initialised with samples from the embedding to perform auto-regressive (dashed line) prediction of output sequences. Middle: A hierarchical auto-encoder with a two-stage latent space. Right: A hierarchical recurrent auto-encoder, the first stage encodes individual time-steps and the second stage embeds series of latent features into a higher-level global latent space.

(i.e. non-sequential) data by combining the ladder auto-encoder structure [267] and variational inference in [246]. Multiple latent codes are learned for a given individual data observation and skip connections are introduced in between hidden layers of the encoder-decoder pairs. These shortcuts directly pass some lower-level information to the decoders and free the upper-level inference from modelling certain local properties of the data. As a result, the ladder architecture enables learning representations that are invariant to different levels of features and allows efficient training of deep VAE models that capture more abstract data structures as the number of latent spaces grows. Another enhancement of the VAE framework is proposed in the two-stage VAE [50] which analyses the issues of posterior collapse and poor ancestral sampling quality which may arise in the trade-off between learning an accurate data representation and achieving a low divergence with the Gaussian prior. The method proposes to pre-train a regular VAE (parameters θ^1, ϕ^1) and use it to encode the whole dataset as $\mathcal{Z} = \{\mathbf{z}^{i=1\dots N}\}$ which gives an approximate manifold $q_{\phi^1}(\mathbf{z}) = \mathbb{E}_{\mathcal{X}} q_{\phi^1}(\mathbf{z}|\mathbf{x})$. In the non-degenerate case, it is likely that $q_{\phi^1}(\mathbf{z}) \neq \mathcal{N}(\mathbf{z}; \mathbb{O}, \mathbb{I})$ which prevents from an accurate ancestral sampling. In the second stage, another VAE (parameters θ^2, ϕ^2) is trained over the dataset encoded by the first VAE and learns the latent representation $\mathbf{z}_2 \sim q_{\phi^2}(\mathbf{z}_2|\mathbf{z})$ of the manifold such that $q_{\phi^2}(\mathbf{z}_2) \approx \mathcal{N}(\mathbf{z}_2; \mathbb{O}, \mathbb{I})$. A two-stage ancestral sampling can then be performed

as:

$$\begin{aligned} \mathbf{z}_2 &\sim \mathcal{N}(\mathbf{z}_2; \mathbb{O}, \mathbb{I}) \\ \mathbf{z} &\sim p_{\theta^2}(\mathbf{z}|\mathbf{z}_2) \\ \mathbf{x} &\sim p_{\theta^1}(\mathbf{x}|\mathbf{z}). \end{aligned} \tag{34}$$

Exact density estimation with normalizing flows. The VAE framework relies on approximate posterior inference, which gives some flexibility over the neural network parameterisation of the latent distribution but prevents direct optimisation of the data likelihood (bounded estimator such as ELBO). This highlights the trade-off between the simplicity of the parametric posterior (e.g. a compressed isotropic Gaussian) and the accuracy of the data density estimation. The Normalizing Flows (NF) framework has been developed in order to perform exact inference and direct optimisation of the data likelihood with invertible neural networks [56]. Whereas the auto-encoder model is comprised of two separate networks which approximately learn to invert each other (such that $\mathbf{x} \approx G_\theta(E_\phi(\mathbf{x}))$), the NF model is based on a bijective network which learns both inference and sampling. A flow is defined as a transformation between two random variables $f_\theta : \mathbf{x} \in \mathbb{R}^d \rightarrow \mathbf{z} \in \mathbb{R}^d$ with an inverse $f_\theta^{-1} = g_\theta$ and according to the change of variable formula the probability distributions can be written as:

$$\log(p_\theta(\mathbf{x})) = \log(p_\theta(\mathbf{z})) + \log \left| det \left(\frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \log(p_\theta(\mathbf{z})) + \log \left| det \left(\mathbf{J}(f_\theta) \right) \right|. \tag{35}$$

Normalizing flows are built by chaining multiples flows in order to allow for complex transformations such that $f_\theta = f_{\theta^1} \circ \ldots \circ f_{\theta^K}$ and choosing a simple and tractable prior such as $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbb{O}, \mathbb{I})$. In this setting the data log-likelihood can be directly optimised as:

$$\log(p_\theta(\mathbf{x})) \propto -\frac{\mathbf{z}^T \mathbf{z}}{2} + \sum_{i=1}^{K} \log \left| det \left(\mathbf{J}(f_{\theta^i}) \right) \right|. \tag{36}$$

Exact samples from this distribution can then be drawn from the prior by using the inverse transform sampling $\mathbf{x} = g_\theta(\mathbf{z})$. The NF framework offers the advantage of an exact density estimation but comes with the limitation that the mapping between data and latents must preserve the dimensionality (no compression). Moreover, as seen in equation 36, training requires using transformations with an efficient computation of the determinant of the Jacobian $det(\mathbf{J}(f_{\theta^i}))$ which has led to the development of several types of flow layers [150] aimed at maximising the expressivity of the transformation

with minimal computation cost of the Jacobian. One of the most commonly used flow layers is the affine coupling layer [57] which splits its input in two parts and use an arbitrary function (e.g. a neural network NN_θ) to predict conditional affine parameters as:

$$\begin{aligned} \mathbf{x}_a, \mathbf{x}_b &= split(\mathbf{x}) \\ \mathbf{s}, \mathbf{t} &= NN_\theta(\mathbf{x}_b) \\ \mathbf{z} &= concatenate(\mathbf{s} \odot \mathbf{x}_a + \mathbf{t}, \mathbf{x}_b). \end{aligned} \tag{37}$$

Because of the identity function applied to $\mathbf{x}_b$, the Jacobian computation of the affine coupling layer is efficiently reduced to $\mathbf{J}(f_\theta) = \log(|\mathbf{s}|)$ while allowing the use of arbitrary neural networks into generative flows for complex data distributions such as natural images [144]. Another application of normalizing flows is proposed in [220] for improved posterior inference in the VAE framework. In this setting, the inference network predicts additional parameters $\lambda_\phi(\mathbf{x})$ of a normalizing flow applied to the usual isotropic Gaussian variational posterior $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \sigma^2_\phi(\mathbf{x}))$. In that sense, the normalizing flows allow a more expressive posterior distribution (Figure 15) to feed the decoder with samples from $q_\phi(\mathbf{z}^K|\mathbf{x}) = f_{\lambda^K_\phi(\mathbf{x})} \circ \ldots \circ f_{\lambda^1_\phi(\mathbf{x})}(q_\phi(\mathbf{z}|\mathbf{x}))$ without sacrificing the convenience of amortised inference with an isotropic Gaussian encoder.

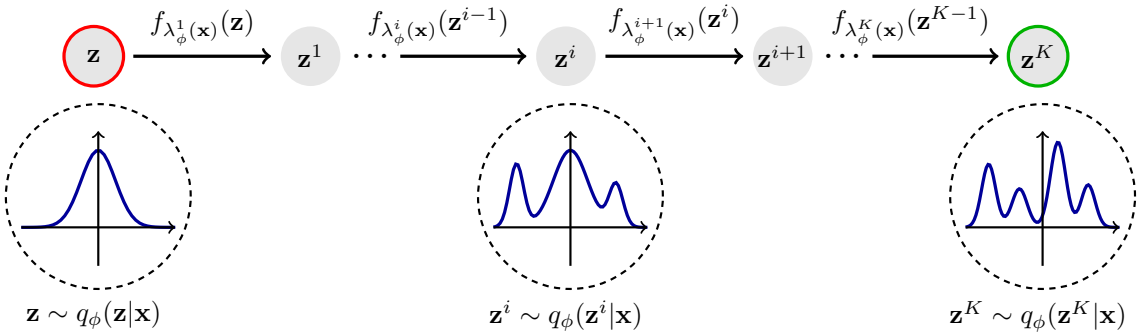

Figure 15: Combining the Normalizing Flows and Variational Auto-Encoder frameworks for enhanced posterior inference[3]*. The regular mean field Gaussian posterior $q_\phi(\mathbf{z}|\mathbf{x})$ can be transformed into a richer distribution with a chain of normalizing flows with parameters predicted by the inference network.*

[3]Figure adapted from a source code.

3.2.3 Implicit density estimation

Generative modelling requires defining a criterion by which the model can learn to approximate the distribution of the data. The previous approaches have mostly relied on maximising the data likelihood under the model constraints such as approximate posterior inference (bounded estimate) or bijective inference (exact estimate). The seminal work on Generative Adversarial Networks (GANs [95]) proposes a totally different approach to learn a data distribution by comparison in a min-max fashion. Adversarial learning relies on a generator-discriminator network pair with competitive optimisation, the discriminator objective is to distinguish between real data samples and generated ones whereas the generator aims at fooling the discriminator. As the learning progresses, the generator G_θ improves the realisticness of its outputs and the discriminator D_ψ refines its ability to separate the generator distribution $p_\theta(\mathbf{x})$ from that of the dataset $p_\mathcal{X}$. Each iteration of this min-max game is defined as:

$$\max_{\psi} \min_{\theta} \mathbb{E}_{p_\mathcal{X}} \log(D_\psi(\mathbf{x})) + \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z})} \log(1 - D_\psi(G_\theta(\mathbf{z}))) \tag{38}$$

and a prior distribution such as $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbb{O}, \mathbb{I})$ is usually chosen and deterministically mapped to the data space by the generator $G_\theta : \mathbf{z} \to \mathbf{x}$. In contrast with the latent variable models previously discussed, the original GAN model does not rely on any inference mechanism which would relate the latent representation to the underlying dataset structure. The optimisation of equation 38 is usually carried in two steps, first the generator parameters are kept fixed and the discriminator update aims at maximising the objective via $D_\psi(\mathbf{x}) \to 1$ (real) and $D_\psi(G_\theta(\mathbf{z})) \to 0$ (fake), second the discriminator parameters are kept fixed and the generator update aims at minimising the objective via $D_\psi(G_\theta(\mathbf{z})) \to 1$ (i.e. fooling the discriminator by being classified as the real data). For a fixed generator, the optimal discriminator distribution is $p^*_\psi(y|\mathbf{x}) = \frac{p_\mathcal{X}}{p_\mathcal{X}+p_\theta(\mathbf{x})}$, with y being the real or fake label predicted by the discriminator. As the generator aims at fitting the dataset as $p_\theta(\mathbf{x}) \approx p_\mathcal{X}$, the min-max training would ideally converge to an equilibrium $p_\psi(y|\mathbf{x}) = 0.5$ regardless of whether samples are drawn from the dataset or generated. Given the optimal discriminator D^*_ψ, the generator minimises the objective

of equation 38 as:

$$\begin{aligned}
\mathcal{L}(\theta|\psi^*) &= \mathbb{E}_{p_{\mathcal{X}}} \log(D^*_\psi(\mathbf{x})) + \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z})} \log(1 - D^*_\psi(G_\theta(\mathbf{z}))) \\
&= \mathbb{E}_{p_{\mathcal{X}}} \log(D^*_\psi(\mathbf{x})) + \mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x})} \log(1 - D^*_\psi(\mathbf{x})) \\
&= \mathbb{E}_{p_{\mathcal{X}}} \log\left(\frac{p_{\mathcal{X}}}{p_{\mathcal{X}} + p_\theta(\mathbf{x})}\right) + \mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x})} \log\left(\frac{p_\theta(\mathbf{x})}{p_{\mathcal{X}} + p_\theta(\mathbf{x})}\right) \\
&= -\log(4) + \underbrace{\mathcal{D}_{KL}(p_{\mathcal{X}}||(p_{\mathcal{X}} + p_\theta(\mathbf{x}))/2) + \mathcal{D}_{KL}(p_\theta(\mathbf{x}))||(p_{\mathcal{X}} + p_\theta(\mathbf{x}))/2)}_{2*\mathcal{D}_{JS}(p_\theta(\mathbf{x})||p_{\mathcal{X}})}.
\end{aligned} \tag{39}$$

This equation satisfies the generator optimum $p^*_\theta(\mathbf{x}) = p_{\mathcal{X}}$ at which $\mathcal{L}(\theta^*|\psi^*) = -\log(4)$ and in this setting it is observed that the GAN algorithm has minimised the Jensen-Shannon divergence $\mathcal{D}_{JS}$ between the dataset distribution and the generator output distribution. In practice, both the generator and the discriminator parameters can be optimized by gradient descent of two losses that update one network parameters while keeping the other fixed:

$$\begin{aligned}
\mathcal{L}(\psi|\theta) &= -\log(D_\psi(\mathbf{x} \sim p_{\mathcal{X}})) - \log(1 - D_\psi(G_\theta(\mathbf{z}))) \\
\mathcal{L}(\theta|\psi) &= -\log(D_\psi(G_\theta(\mathbf{z}))).
\end{aligned} \tag{40}$$

In the GAN framework the metric is learned throughout the training rather than fixed a priori, this technique has enabled unprecedented performances in the domain of computer vision and image generation [283] (e.g. photo-realistic image synthesis). Adversarial metric learning has also proven effective in other settings of distribution matching, this includes posterior inference regularisation [173] (e.g. replacement of analytical latent divergences) and novel tasks related to unpaired domain conversion [3] (no metric a priori). However, adversarial training suffers from instability [46] and GANs are prone to mode collapse [301] which happens when the generator focuses on a single mode of the data distribution and fails at producing diverse samples. These known issues with GANs as well as the ongoing quest for increased sampling quality (e.g. generating larger images and higher resolutions) have motivated an intensive research which comprises both alternative model formulations and improved architecture design [49].

Methods relying on two-sample tests have been proposed amongst alternative formulations of the implicit data distribution matching task. The two-sample test mea-

sures how dissimilar are the distributions of two independent batches of samples and can be analytically computed with the Maximum Mean Discrepancy (MMD [97]). In comparison with GANs, it may be considered as a fixed discriminator that a generator tries to fool in order to produce samples that are statistically indistinguishable from the dataset [67]. The discriminative power of the MMD against the distribution complexity depends on the choice of kernel for density estimation, the amount of data to compute the statistics and the dimensionality of the observed space. While training a generator with a MMD-based objective does not suffer from the GAN instability, it as well does not compete with the expressivity of adversarial learning (i.e. the GAN discriminator can provide a finer loss than the fixed kernel MMD), which has motivated some further investigations. A parametric MMD kernel can be trained [161] in order to refine standard kernels such as radial basis and inverse multi-quadratic functions while having much fewer parameters than a traditional discriminator network. Another work proposes to train an unregularised auto-encoder which may reach a satisfying reconstruction quality, and to use MMD to train a generator to match its embedding statistics [162]. While unregularised auto-encoders can achieve accurate reconstructions, they do not guarantee good sampling. By computing the two-sample test in this compressed embedding, distribution matching may be more efficient than in the data space and conveniently lets the generator benefit from the auto-encode pretraining.

Adversarial auto-encoders. The training of GANs relies on a network pair, a generator and an auxiliary discriminator that is used for training and often discarded once the generator has converged to a sufficient performance. In contrast with auto-encoders, the vanilla GAN does not have an inference mechanism which allows meaningful interactions with the latent representation (e.g. visualising the latent distribution of the dataset) and can support down-stream tasks (e.g. encoder as unsupervised feature extractor). In order to broaden the use of GANs beyond their success at the generative task, some techniques have been proposed to inverse a pre-trained generator [48] [159] via gradient descent. Given an observation $\mathbf{x}$, the inversion aims at finding its latent representation $\mathbf{z}^*$ such that $G_\theta(\mathbf{z}^*)$ is the closest match to the observation. To this extent, a random latent can be sampled from the prior $\mathbf{z} \sim p_\theta(\mathbf{z})$ and iteratively updated by optimising:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} ||\mathbf{x} - G_\theta(\mathbf{z})||_2. \tag{41}$$

In the case of a deterministic generator each latent is mapped to a unique data sample, however there may be multiple latents that approximately correspond to the inverse of a given observation. Because these inversion techniques are iterative and inherently ubiquitous, their application to approximate an inference mechanism is inefficient. On the other hand, several models have been proposed in order to complement the auto-encoder framework with the strengths of adversarial learning. A VAE/GAN model is proposed in [157] which replaces the data-likelihood objective of the VAE with the GAN objective (enforcing realistic output samples) and a reconstruction cost computed in the hidden activations of the discriminator. Whereas the distance computed in the pixel space does not yield a good metric of perceptual similarity (e.g. it is not translation invariant), the distance computed in the hidden layers of the discriminator is expected to give a better reconstruction metric which spans observation patches of multiple sizes and improves in the course of the adversarial training. The GAN framework has also been applied to distribution matching in the latent space of an auto-encoder, using a discriminator as a probabilistic regularizer of the aggregated posterior inference. The Adversarial Auto-Encoder (AAE [174]), which is an instance of Wasserstein Auto-Encoder (WAE-GAN [263]), replaces the KL divergence with a divergence $\mathcal{D}_{GAN}$ learned in an adversarial fashion:

$$\begin{gathered}\max_{\psi}\min_{\phi}\ \underbrace{\mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z})}\log(D_\psi(\mathbf{z}))}_{\text{classifier on the prior}}+\underbrace{\mathbb{E}_{p_\mathcal{X}}\log(1-D_\psi(E_\phi(\mathbf{x})))}_{\text{classifier on the posterior}}\equiv\mathcal{D}_{GAN}(E_\phi(\mathbf{x})||p_\theta(\mathbf{z}))\\ \psi^*,\theta^*,\phi^*=\max_{\psi}\min_{\theta,\phi}\ -\mathbb{E}_{\mathbf{z}\sim E_\phi(\mathbf{x})}\log(p_\theta(\mathbf{x}|\mathbf{z}))+\mathcal{D}_{GAN}(E_\phi(\mathbf{x})||p_\theta(\mathbf{z})).\end{gathered}\tag{42}$$

to approximate the Jensen-Shannon divergence $\mathcal{D}_{JS}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))$. Akin to the WAE-MMD, the AAE does not require a probabilistic encoder and can allow any prior and any posterior parametrisation. Moreover, adversarial learning in the latent space is potentially more stable than in the data space since latents usually have a reduced number of dimensions and simpler distributions than that of the data observations.

Both the VAE/GAN and the AAE make use of the GAN framework in order to learn the latent variable model $p(\mathbf{x},\mathbf{z})$, either via adversarial distribution matching in the data space (VAE/GAN) or as a latent regularizer (AAE). In the auto-encoder setting, this amounts to learn the two conditional distributions $q_\phi(\mathbf{z}|\mathbf{x})$ (inference with an encoder E_ϕ) and $p_\theta(\mathbf{x}|\mathbf{z})$ (generation with a decoder G_θ) to approximate the joint

LVM probability as:

$$\begin{aligned} q_\phi(\mathbf{x}, \mathbf{z}) &= q_\phi(\mathbf{z}|\mathbf{x}) q_\phi(\mathbf{x}) \\ p_\theta(\mathbf{x}, \mathbf{z}) &= p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}) \end{aligned} \tag{43}$$

and the marginal distributions are commonly fixed as $q_\phi(\mathbf{x}) \equiv p_\mathcal{X}$ (the observed dataset) and $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}, \mathbb{O}, \mathbb{I})$ (fixed prior). From this perspective, a LVM can be learn via adversarial distribution matching of the joint distributions which leads to the Adversarially Learned Inference (ALI [64], also proposed as Bi-GAN [60]) which solely relies on the GAN framework for both inference and generation. The task of joint distribution matching is implemented with a discriminator $D_\psi(\mathbf{x}, \mathbf{z})$ that learns to classify pairs of data and latent samples $p_\psi(y|\mathbf{x}, \mathbf{z})$ and a min-max game in which the encoder-decoder pair competes with the discriminator as:

$$\max_\psi \min_{\theta,\phi} \mathbb{E}_{p_\mathcal{X}} \log(D_\psi(\mathbf{x}, E_\phi(\mathbf{x}))) + \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z})} \log(1 - D_\psi(G_\theta(\mathbf{z}), \mathbf{z})). \tag{44}$$

This objective jointly matches $E_\phi : \mathbf{x} \to \mathbf{z}$ to the latent prior and $G_\theta : \mathbf{z} \to \mathbf{x}$ to the dataset observations, leading to an optimum with E_ϕ and G_θ inverting each other as expected in the auto-encoder setting.

The common goal of generative modelling frameworks is to approximate the underlying distribution of a dataset in order to sample new data which is consistent with the training observations (e.g. realistic in the sense of GANs) and diverse. With respect to the ancestral sampling in latent variable models, the first property amounts to generating data with high-likelihood from any latent sample of the prior and the second property amounts to a conditional generative distribution that covers all the modes observed in the dataset. Nonetheless other generative qualities may be desired such as disentanglement, separating the factors of variation onto separate and interpretable latent dimensions, and interpolation smoothness, the ability to continuously morph the semantic characteristics of two samples. As shown in [111], the latent representation of a Variational Auto-Encoder with a larger β weighting of the KL divergence tends to have an increased disentanglement and smoothness. This trend is however traded-off by a lower reconstruction quality, which traduces in more blurry samples. A variant of the GAN framework is proposed in the Adversarially Constrained Auto-encoder Interpolation (ACAI [15]) for improving the interpolation quality of an auto-encoder without impeding its reconstruction quality. An interpolation is defined as a convex combination of two latent codes $\mathbf{z}_\alpha = \alpha E_\phi(\mathbf{x}_1) + (1 - \alpha) E_\phi(\mathbf{x}_2)$ and a desired quality is

that the generated interpolations $\mathbf{x}_\alpha = G_\theta(\mathbf{z}_\alpha)$ smoothly morph between the semantic characteristics of $\mathbf{x}_1$ and $\mathbf{x}_2$ over $\alpha : 1 \rightarrow 0$ while remaining perceptually consistent. This property is encouraged by complementing the reconstruction objective of an auto-encoder (e.g. MSE on data observations) with a separate adversarial regularizer that is applied to random interpolations. The discriminator is trained at predicting the mixing coefficient $D_\psi : \mathbf{x}_\alpha \rightarrow \alpha$ with random $\alpha \in [0, 0.5]$ and the auto-encoder competes at being classified as $\alpha = 0$. The ACAI training is thus minimising the following two losses:

$$\begin{aligned} \mathcal{L}_\psi &= \underbrace{||D_\psi(\mathbf{x}_\alpha) - \alpha||^2}_{\text{classification}} + \underbrace{||D_\psi(\gamma\mathbf{x} + (1-\gamma)G_\theta(E_\phi(\mathbf{x})))||^2}_{\text{discriminator regularisation}} \\ \mathcal{L}_{\theta,\phi} &= \underbrace{||\mathbf{x} - G_\theta(E_\phi(\mathbf{x}))||^2}_{\text{reconstruction}} + \lambda \quad \underbrace{||D_\psi(\mathbf{x}_\alpha)||^2}_{\substack{\text{adversarial interpolation} \\ \text{regularizer}}} \end{aligned} \tag{45}$$

with hyper-parameters λ, γ and a regularisation term in $\mathcal{L}_\psi$ that forces the discriminator to output zero for non-interpolated data and lets it analyse ground-truth observations $\mathbf{x}$.

Unsupervised domain translation. The GAN framework is a highly flexible approach to unsupervised distribution matching and has opened new directions in generative modelling for domain translation. An extended body of research has been carried in image conversion [3] which is predominantly performed in the unsupervised setting that does not assume paired data across domains, although it can as well use paired data in a conditional GAN setting [129]. Given two or more data domains $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_P$, the task consists in learning a model that would translate a source sample $\mathbf{x}_i \in \mathcal{X}_i \; \forall i \in [1, P]$ into a sample $\hat{\mathbf{x}}_j$ perceived as belonging to another target domain $\mathcal{X}_j \; \forall j \in [1, P] \neq i$. While the output should be aligned with the target domain, it should as well preserve the source content and remain structurally consistent. One such example would be an image translation which converts in between photographs and paintings [304] (e.g. converting a landscape picture into a Van Gogh stylised artwork). To this extent, it should be noted that there often is no underlying ground-truth of what is the expected translation output (e.g. Van Gogh has not painted that landscape), which justifies the need for unsupervised methods to learn from unpaired datasets. Moreover domain translation is often ill-defined and requires a strong degree of extrapolation. Unpaired data is thus suited as it gives the model a large degree of freedom at learning correspondences beyond biases of human annotations.

Image stylisation was introduced in the seminal work on style transfer [88] as the task of combining features from a source sample $\mathbf{x}_s$ with features from another target sample $\mathbf{x}_t$. These features are extracted at different semantic levels using the hidden activations of the pretrained VGG [243] large-scale image classifier, its lower layers extract local patterns and its upper layers have a larger receptive field which can capture more abstract features. A randomly initialised output image $\mathbf{x}_o$ is then iteratively optimised by minimising a content loss and a style loss. The content loss applies the source structure to the output by minimising the distance between their hidden layer activations VGG^l_{ik}, considering the l-th convolutional layer it gives:

$$\mathcal{L}_{\text{content}}(\mathbf{x}_s, \mathbf{x}_o, l) = \frac{1}{2} \sum_{i,k} (\text{VGG}^l_{ik}(\mathbf{x}_s) - \text{VGG}^l_{ik}(\mathbf{x}_o))^2 \tag{46}$$

with i the channel index and k the spatial position. The style loss applies the target texture to the output by matching their feature statistics computed via the Gram matrice across channels i, j:

$$\begin{aligned} \text{Gram}_{ij}(\text{VGG}^l(\mathbf{x})) &= \sum_k \text{VGG}^l_{ik}(\mathbf{x}) \text{VGG}^l_{jk}(\mathbf{x}) \\ \mathcal{L}_{\text{style}}(\mathbf{x}_t, \mathbf{x}_o, l) &= \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (\text{Gram}_{ij}(\text{VGG}^l(\mathbf{x}_t)) - \text{Gram}_{ij}(\text{VGG}^l(\mathbf{x}_o)))^2 \end{aligned} \tag{47}$$

with N_l the number of channels and M_l the output size of the l-th convolution layer. Each optimisation step is then minimising the gradient of a weighted sum of both the content and style losses across various hidden layers with respect to the output image being generated. In this setting the conversion is performed by combining the spatial features of the source sample, the content, with the texture features (local statistics) of the target sample, the style. It should be noted that the technique does not involve training any generative model, that it only applies to individual samples rather than datasets and that the optimisation parameters are empirically set (e.g. which hidden layers of the pretrained classifier).

While this approach to neural style transfer has motivated several follow-up researches [132], GAN-based methods have been developed at the scale of datasets (defining each domain) which allow to train dedicated models rather than using features of pretrained classifiers and hand-tuned heuristics. In the unpaired setting, a common approach is to use the GAN algorithm with the discriminator(s) forcing the generator(s)

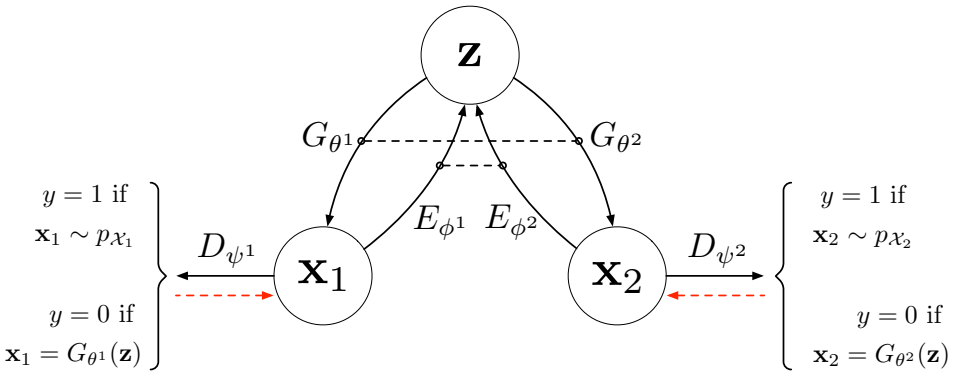

Figure 16: The Unsupervised Image-to-Image Translation Networks model with a bi-directional VAE/GAN and a shared latent space. Black dashed connections denote the weight sharing between the upper layers of the encoder/decoder pairs.

to output a realistic distribution for each of the target domains (domain-specific features or style). Some additional constraints (e.g. weight-sharing, cycle-consistency) are usually applied in order to learn a meaningful mapping which preserves the semantic content (domain-independent features or structure) of a particular source sample. The Cycle-GAN approach [304] is a bi-directional translation model with two generator and discriminator pairs such as $G_{\theta^{1\to2}}$ and D_{ψ^2} (translate to and classify in $\mathcal{X}_2$). A cycle-consistency regularisation loss is added to the GAN training which encourages the model to learn a bijective mapping between the two domains such that $G_{\theta^{2\to1}}(G_{\theta^{1\to2}}(\mathbf{x}_1)) \approx \mathbf{x}_1$. The training of one domain translation and its inverse cycle-consistency thus follows:

$$\begin{aligned}\max_{\psi^2} \min_{\substack{\theta^{1\to2} \\ \theta^{2\to1}}} & \mathbb{E}_{p_{\mathcal{X}_2}} \log(D_{\psi^2}(\mathbf{x}_2)) + \mathbb{E}_{p_{\mathcal{X}_1}} \log(1 - D_{\psi^2}(G_{\theta^{1\to2}}(\mathbf{x}_1))) \\ & + ||G_{\theta^{2\to1}}(G_{\theta^{1\to2}}(\mathbf{x}_1)) - \mathbf{x}_1||_1\end{aligned} \tag{48}$$

and the whole model jointly optimises both domain-dependent GAN objectives and their corresponding cycle-consistent inverse regularisations. The Unsupervised Image-to-Image Translation Networks (UNIT [164]) introduce cycle-consistency across the latent space of a bi-directional VAE with a weight-sharing constraint [165] in the upper layers and domain-specific discriminators (Figure 16). Similar to Cycle-GAN, each discriminator ensures that the decoder outputs match their respective domain distributions but the regularisation and weight-sharing strategies enable the learning of a shared latent space which encodes domain-invariant features. This follows the hypothesis that matched data observations $\{\mathbf{x}_1^*, \mathbf{x}_2^*\}$ would have a common latent representation $\mathbf{z}^* = E_{\phi^1}(\mathbf{x}_1^*) = E_{\phi^2}(\mathbf{x}_2^*)$ which would equally allow recovering them as $G_{\theta^1}(\mathbf{z}^*) = \mathbf{x}_1^*$

and $G_{\theta^2}(\mathbf{z}^*) = \mathbf{x}_2^*$. In addition to the domain-specific GAN and VAE objectives, cycle-consistency regularisations are added by bi-directional auto-encoding which yields two KL divergences and a likelihood term for input reconstruction. Considering the cycle-consistency from and back to domain $\mathcal{X}_1$ and a shared prior $p_\theta(\mathbf{z})$, the regularisation is computed as:

$$\begin{aligned} \mathbf{x}_{1\to2} &= G_{\theta^2}(E_{\phi^1}(\mathbf{x}_1)) \\ \mathcal{L}_{CC_1} &= \mathcal{D}_{KL}(q_{\phi^1}(\mathbf{z}_1|\mathbf{x}_1)||p_\theta(\mathbf{z})) + \mathcal{D}_{KL}(q_{\phi^2}(\mathbf{z}_2|\mathbf{x}_{1\to2})||p_\theta(\mathbf{z})) \\ &\quad -\mathbb{E}_{\mathbf{z}_2\sim q_{\phi^2}(\mathbf{z}_2|\mathbf{x}_{1\to2})} \log(p_{\theta^1}(\mathbf{x}_1|\mathbf{z}_2)). \end{aligned} \tag{49}$$

One of the motivations of unpaired domain translation stems from the ill-defined nature of the task, meaning that there may be multiple outputs that can plausibly reflect the given input into the target domain. The Multimodal Unsupervised Image-to-Image Translation (MUNIT [122]) extends the previous works to generating multiple target domain instances for a given translation input by learning a disentangled latent space of content, $\mathbf{z}_c$ shared across domains, and style, separate $\mathbf{z}_{s1}, \mathbf{z}_{s2}$ for each domain. This enables a multi-modal translation by encoding a source sample $E_{\phi^1} : \mathbf{x}_1 \to \mathbf{z}_c, \mathbf{z}_{s1}$ and combining its content latent code with random samples of the target style latent space which are decoded as $\mathbf{x}_{1\to2} = G_{\theta^2}(\mathbf{z}_c(\mathbf{x}_1), \mathbf{z}_{s2} \sim \mathcal{N}(\mathbb{O}, \mathbb{I}))$. The disentanglement of the latent space is enforced by additional unsupervised training objectives such that by encoding the translation output $E_{\phi^2}(\mathbf{x}_{1\to2})$ the model retrieves the initial content code $\mathbf{z}_c(\mathbf{x}_{1\to2}) = \mathbf{z}_c(\mathbf{x}_1)$ as well as the randomly sampled style code $\mathbf{z}_{s2}(\mathbf{x}_{1\to2}) = \mathbf{z}_{s2}$ (the same procedure is jointly applied to the translation $\mathbf{x}_{2\to1}$).

The task of bi-directional domain translation is efficiently learned by cycle-consistent GANs under the assumption that the two domains are structurally similar (e.g. texture differences) and homogeneous. However they tend to fail when applied to more disparate domains which have a large structural and semantic gap. The TraVeL-GAN [4] proposes a less constrained training strategy than that of cycle-consistency (i.e. invertibility) which allows it to produce reasonable conversions between domains at which cycle-consistent GANs fail. This uni-directional translation model comprises a generator-discriminator pair as well as a Siamese network [25] which acts like an encoder E_ϕ over both the source and target domains to learn a shared embedding. In this setting the individuality relationship, which output sample should correspond to a given input sample, is learned via a similarity metric in the shared embedding of

the siamese network. Given two inputs $\mathbf{x}_i, \mathbf{x}_j$ and the corresponding generator outputs $G_\theta(\mathbf{x}_i), G_\theta(\mathbf{x}_j)$, two embedding vectors can be computed as $\nu(\mathbf{x}_i, \mathbf{x}_j) = E_\phi(\mathbf{x}_j) - E_\phi(\mathbf{x}_i)$ and $\nu(G_\theta(\mathbf{x}_i), G_\theta(\mathbf{x}_j))$ and compared with the euclidean (magnitude) and cosine (angle) distances. The generator and siamese network jointly minimise this embedding distance under the constraints of the target domain adversarial distribution matching and a margin-based contrastive objective which forces the embedding not to collapse (e.g. zero distance for every latent points). In contrast with cycle-consistency, this embedding distance offers more flexibility since the siamese network can freely learn which features are relevant to compute the cross-domain similarity metric.

Another challenge arises from the definition of the previous methods at performing cross-domain generation with dedicated networks (either uni-directional or bi-directional). These methods require a specific generator-discriminator pair per target domain and thus do not scale efficiently to translations across more than two domains. Multi-domain translation within a single generator-discriminator pair is proposed in Star-GAN [42], which aims at processing an arbitrary number of domains $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_P$ with a fixed architecture. The single discriminator has an auxiliary classification loss which forces it to assess the realisticness of any sample $\mathbf{x}_i \in \mathcal{X}_i \ \forall i \in [1, P]$ while discriminating its domain membership $D_\psi : \mathbf{x}_i \rightarrow y, i$ ($y = 1$ if true, $y = 0$ if generated). Consequently, the single generator is fed with samples from any source domain as well as a label of the target domain to which it generates as $G_\theta : \mathbf{x}_i, j \rightarrow \mathbf{x}_{i \rightarrow j}$. While the discriminator has a classification objective to favour a domain-specific discrimination, the generator has a cycle-consistency objective that requires an accurate output target such that $G_\theta(\mathbf{x}_{i \rightarrow j}, i) \approx \mathbf{x}_i$. This approach to unpaired domain translation attractively scales to many-to-many domain generation, on the one hand it could be traded-off with some loss of domain-specific accuracy and diversity, or it may as well benefit from more data and some overall modelling improvements by multi-task learning [43].

3.2.4 Model conditioning

The aforementioned methods of domain translation have opened novel avenues of generation, which is mainly driven by example. By providing a source example and a target domain, one can implicitly specify an input content and a desired output style variation. The extension to multi-domain translation [42] could allow a finer granularity of control by specifying sub-domains of a given database, for instance a collection of human faces

[166] may be divided into domains corresponding to annotated gender, age, hair and skin colours. Nonetheless, such definition of domain-specific features is somehow arbitrary since there are inherent overlaps in between classes (e.g. a blond haired person can have any gender). In this regard, the use of a single network to generate multiple target attributes raises several issues on how to model domain-invariant features and how to effectively provide the conditioning information (as opposed to training separate networks for each target). An effective learning of conditional generation should provide a model that accurately generalises to unseen pairs of input and target label, moreover it could allow the regression of continuous output attributes [55] (as opposed to categorical domain classes) with potential feature interpolations.

In comparison with the unsupervised distribution matching task which aims at capturing all modes of a dataset at once, conditional generation aims at learning many per-class distributions. Star-GAN uses a conditional generator with one-hot encoded label concatenation at its input and a discriminator with an auxiliary classification objective. Besides the somehow ill-defined nature of the classification objective for multi-attribute learning, it can be observed that the conditioning mechanism at the generator input may not be optimal. In a generic conditional GAN [194] (Figure 18), the feedforward generation proceeds from a randomly sampled prior vector of low dimension which is passed through several layers with up-sampling, the overall semantic structure (global features) is shaped by input layers while those by the output process a smaller receptive field (local features). Class-specific properties may be perceived in both the global structure and the local statistics of data, given that the generator is provided the conditioning information at its input, it should learn to pass that information throughout the hidden layers in order to allow the ultimate layer to equally produce conditional features as the input layer does. As an example, an image generator conditioned on animal classes such as birds, fishes, horses and zebras should both generate some consistent body shapes (e.g. large animal with four legs) and textures (e.g. zebra stripes or not). From this perspective, one key aspect of conditioning lies in the method by which hidden features are altered by the conditioning information. One direct modification to the input concatenation is to perform label concatenation for all hidden layers, which amounts to learning a conditional biasing of hidden features [65]. Another approach is conditional scaling, which replaces the additive conditioning by a multiplication, or gating if using a sigmoidal scaling activation that selectively filters hidden activations. In this spirit, the Feature-wise Linear Modulation (FiLM [204])

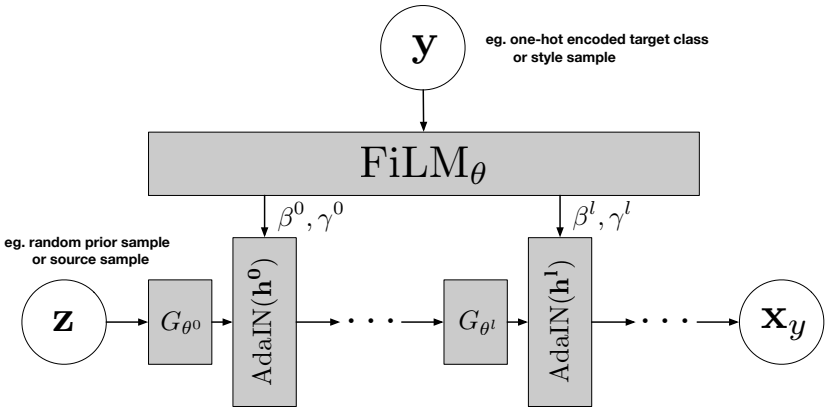

Figure 17: Overview of a generative model using Adaptive Instance Normalisation (AdaIN) with conditional biasing and scaling parameters computed with FiLM layers. The statistics of each hidden layer activations are first normalised and then modulated with conditional parameters extracted from a target class representation or a style sample. This approach lets the model (generator+FiLM) adaptively apply conditioning information at different steps of the computation (from higher-level at the input to lower-level at the output)

.

proposes a general purpose conditioning layer which combines conditional biasing and scaling. To this extent we consider a generator $G_\theta : \mathbf{z}, \mathbf{y} \rightarrow \mathbf{x}_y$ with $\mathbf{y}$ some representation of the conditioning information, such as a one-hot vector of the class y or a target style sample (Figure 17). For generality we refer to the input as a latent $\mathbf{z}$, although in domain translation it would correspond to a source sample of another domain (e.g. $G_\theta : \mathbf{x}_{\neq y}, \mathbf{y} \rightarrow \mathbf{x}_y$). Given $\mathbf{h}^l(\mathbf{z})$ the activations of any l-th hidden layer, a FiLM layer produces conditional bias $\beta^l_\theta(\mathbf{y})$ and scale $\gamma^l_\theta(\mathbf{y})$ parameters which are applied as:

$$\text{FiLM}(\mathbf{h}^l(\mathbf{z})) = \gamma^l_\theta(\mathbf{y}) \odot \mathbf{h}^l(\mathbf{z}) + \beta^l_\theta(\mathbf{y}) \tag{50}$$

before feeding the subsequent hidden layer. This technique modifies the hidden feature statistics, which resembles the process of the many normalisation techniques proposed in the literature. One broadly used normalisation technique is Batch-Normalisation (BN [127]) which accelerates neural network training by normalising the mean and standard deviation for each individual feature dimension (e.g. channel of a convolution) as:

$$\text{BN}(\mathbf{h}^l(\mathbf{z})) = \gamma \odot \left(\frac{\mathbf{h}^l(\mathbf{z}) - \mu(\mathbf{h}^l)}{\sigma(\mathbf{h}^l)} \right) + \beta \tag{51}$$

with γ, β two learned affine parameters and $\mu(\mathbf{h}^l), \sigma(\mathbf{h}^l)$ the mean and standard deviation of the l-th layer activations computed over the batch size. An alternative nor-

malisation technique called Instance Normalisation (IN) is introduced for stylisation [265] which computes statistics $\mu(\mathbf{h}^l), \sigma(\mathbf{h}^l)$ across spatial dimensions independently for each sample (and each channel if applied to a convolution). Since the IN statistics are individually computed over the spatial dimensions, it is responsible for contrast normalisation and may be interpreted as a style normalisation [121] which would whiten the Gram matrice used in style transfer [88]. Several models [121] [66] [90] have used this property for Adaptive Instance Normalisation (AdaIN) which uses conditional affine parameters to alter the instance normalised activations, for instance extending the FiLM framework as:

$$\text{AdaIN}(\mathbf{h}^l(\mathbf{z})) = \gamma^l_\theta(\mathbf{y}) \odot \underbrace{\left(\frac{\mathbf{h}^l(\mathbf{z}) - \mu(\mathbf{h}^l)}{\sigma(\mathbf{h}^l)}\right)}_{\text{IN statistics}} + \beta^l_\theta(\mathbf{y}). \tag{52}$$

Feature normalisation and conditional re-scaling are efficiently applied to feed-forward generation and tasks such as image-to-image translation and style transfer, by removing some domain-specific information and shifting feature statistics to match that of a target. Specific frameworks have been introduced in the auto-encoder setting, which address the issue of disentangling attribute conditions from the latent space [103] [156]. A conditional auto-encoder is usually comprised of an unsupervised encoder, an analysis that does not require input labels, and a conditional decoder, a generator which is provided supervision in the forms of controls. In that case the encoder can freely extract any features from the data, including the attributes of the conditioning. This would lead to a decoder that can ignore its conditioning by relying solely on the encoded features, thus making conditioning ineffective at test time. These observations have motivated the learning of an attribute-invariant latent representation in the Fader Networks [156] which uses an adversarial regularizer to constrain the encoder representation to be disentangled from attribute conditions (Figure 18). This approach is reminiscent of domain adaptation techniques [86] [288] which rely on making predictions based on features that are invariant between source and target domains in order to allow efficient knowledge transfer for classification in the unlabelled target domain. Given data $\mathbf{x}$ paired with K ground-truth binary attributes $\mathbf{y} = \{0, 1\}^K$, adversarial regularisation in the Fader Networks is enforced by a latent discriminator trained at classifying $D_\psi : E_\phi(\mathbf{x}_y) \rightarrow \mathbf{y}$ while the encoder is trained such that the discriminator fails and predicts the opposite labels $\bar{\mathbf{y}}$. As a result, the decoder should retrieve the in-

put data $G_\theta : \mathbf{z}, \mathbf{y} \rightarrow \mathbf{x}_y$ by making use of both the attribute-invariant code $\mathbf{z} \equiv E_\phi(\mathbf{x}_y)$ and the ground-truth attributes. The overall training is done minimising the two opposite adversarial objectives (NLL classification) as well as the reconstruction loss (MSE):

$$\begin{aligned} \mathcal{L}_{\text{disc.}}(\psi|\phi) &= -\log(p_\psi(\mathbf{y}|E_\phi(\mathbf{x}_y))) \\ \mathcal{L}_{\text{ae.}}(\theta, \phi|\psi) &= ||G_\theta(E_\phi(\mathbf{x}_y), \mathbf{y}) - \mathbf{x}_y||_2^2 - \log(p_\psi(\bar{\mathbf{y}}|E_\phi(\mathbf{x}_y))). \end{aligned} \tag{53}$$

This approach has several advantages, it uses adversarial training in the compressed latent space rather than in the observation space, it readily applies to multiple attributes and categorical feature distributions (e.g. multiple hair colours). Moreover the authors report that feature interpolation may be performed using y as a fader, or mixing coefficient, such that setting the hair attribute half-way between blond and black would make brown colour. The underlying idea is that two hypothetical data pairs, matched up to the attribute, would correspond to the same latent and could be equally decoded.

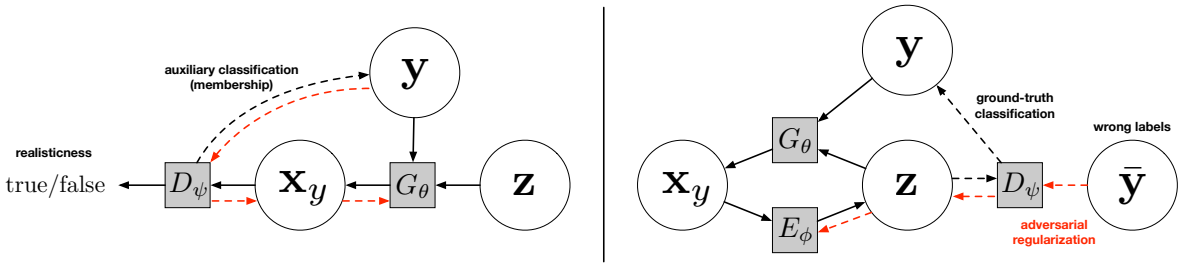

Figure 18: Left: A conditional generator is trained in a GAN fashion with a discriminator evaluating the realisticness of the synthetic samples against data observations, as well as the class membership via an auxiliary classification loss. Right: A Fader Networks autoencoder learns an attribute invariant representation with an adversarial latent classifier. The attributes conditioning the decoder are disentangled from the encoder representation by an adversarial regularisation. The classifier is trained at inferring the ground-truth attributes while the decoder competes at being classified wrongly.

3.2.5 Learning with perceptual embeddings

In the first place, the training of a generative model aims at fitting its parametric output distribution to the observed distribution of a dataset. This may be done with an explicit probabilistic formulation such as maximising the likelihood of the model with respect to each ground truth sample, or with a deterministic regression such as minimising

some reconstruction error in the observation space (e.g. pixel-wise MSE). It can also be done with an implicit formulation which matches the expectation of the model output with samples randomly drawn from the dataset, usually following the GAN framework. One main drawback of the first approach would be that distances in the pixel space are little aligned with perception (e.g. no shift invariance) and may not provide the model with insights on the semantic structure of the observations (global features). As detailed in the previous section, these models may be enhanced by learning a latent space which embeds a global representation of the data. The second approach takes advantage from learning the metric with a discriminator that is trained at assessing the distance between the distributions of the model output and the dataset, throughout the generator training. A generic discriminator, akin to an encoder, usually has an architecture mirroring that of the generator and is able to extract features at different semantic levels (e.g. increasing receptive fields) until its output which is reduced to a scalar prediction (real or synthetic, as opposed to latent parameters of an encoder). For these reasons, the discriminator can provide a refined divergence measure which relies on both local and global features. On the other hand, GAN training in the data space is prone to instability and does not actually guarantee a good convergence (e.g. mode collapse when the generator and discriminator only model a limited subset of the data variations). As proposed in the VAE/GAN [157], these two approaches may be combined by using the discriminator to assess the model realisticness as well as computing a reconstruction error in its hidden activation space. This idea, termed "auto-encoding beyond pixels using a learned similarity metric", emphasises the power of computing the reconstruction error in a perceptual embedding rather than in the observation space. The perceptual embedding refers to multi-scale features extracted by a network (Figure 19) trained on an auxiliary task which mimics a process of perception. In this case, it refers to the discriminator task of predicting whether samples are true or synthetic. In the seminal work on neural style transfer [88] [87], feature matching losses are computed in the hidden layers of a pretrained classifier which is another instance of perceptual embedding that results from a high-level recognition task.

Several works have leveraged large-scale models pretrained in the image domain to improve generative modelling with perceptual losses. Besides the usefulness of the deep feature representations [299], it should be noted that they are readily applicable because computed with neural networks which are inherently designed for GPU accelerated and differentiable processing, an efficiency condition which is not always met by hand-

crafted metrics. An image translation model is proposed in [133] which optimises style and content losses computed in the hidden feature space of VGG. For generality, we refer to the l-th layer of the pretrained feature extractor as $F^l(.)$ and do not specify its parameters which are kept fixed. Given a sample $\mathbf{x}_1$ in the source domain and a corresponding sample $\mathbf{x}_2$ in the target domain, a translation model $G_{\theta^{1\to 2}} : \mathbf{x}_1 \to \hat{\mathbf{x}}_2$ is trained similarly to equations 46,47 by minimising losses:

$$\begin{aligned} \mathcal{L}^l_{\text{content}} &\propto ||F^l(\mathbf{x}_1) - F^l(G_{\theta^{1\to 2}}(\mathbf{x}_1))||_2^2 \\ \mathcal{L}^l_{\text{style}} &\propto \sum_{i,j} ||\text{Gram}_{ij}(F^l(\mathbf{x}_2)) - \text{Gram}_{ij}(F^l(G_{\theta^{1\to 2}}(\mathbf{x}_1)))||_F^2. \end{aligned} \tag{54}$$

An alternative model [61] proposes to replace the use of a hand-crafted style loss based on Gram matrices by an adversarial loss in the target domain which is combined with both deep feature and pixel-wise distances. A perceptual distance is also used as an auto-encoder reconstruction loss in [207] which reports representation learning improvements assessed by increased classification performances using the trained latent features as inputs to subsequent predictive tasks. The use of a perceptual embedding is also proposed in [228] as an extension to generative moment matching [67] [162] using feature statistics (e.g. MMD, mean and covariance) within the activations of a pretrained classifier. Another potential use of a pretrained classifier is to promote the accurate generation of specific attributes of the data via a regression loss, which is reminiscent of the auxiliary classification loss in conditional GANs. Assuming a conditional generator $G_\theta : \mathbf{z}, y \to \mathbf{x}_y$ and a classifier pretrained on the task $F : \mathbf{x}_y \to y$, the auxiliary loss may be provided as a regression:

$$\mathcal{L}_{\text{auxiliary}} = |F(G_\theta(\mathbf{z}, y)) - y|. \tag{55}$$

3.3 Evaluation of Generative Modelling

The common methodology of deep learning relies on a given dataset to exemplify and simulate a certain task. An overall model structure defines the information flow and interactions between variables, which is parametrised with neural networks. The randomly initialised space of model parameters is iteratively updated by an optimisation algorithm which performs stochastic gradient descent of a loss function. This training

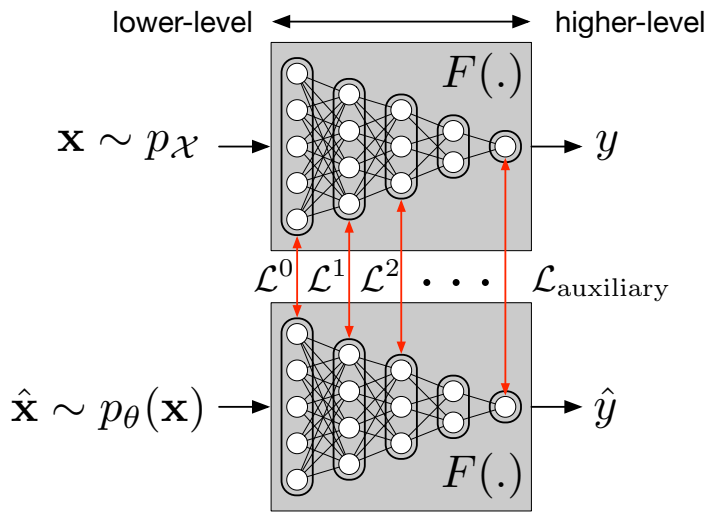

Figure 19: A pretrained feature extractor $F(.)$, e.g. a classifier, can be used as a perceptual embedding to compute feature distances in its hidden activation space. Layers at different depths extract features associated with increasing degrees of abstraction and usually spanning a larger receptive field (e.g. in CNNs), up to the final layer which performs a semantic prediction. Perceptual losses can be used as reconstruction errors for $\hat{\mathbf{x}} \approx \mathbf{x}$ as well as for domain translation by computing losses of $\hat{\mathbf{x}} \equiv \mathbf{x}_{1\to 2}$ with samples from both $p_{\mathcal{X}_1}$ (source loss) and $p_{\mathcal{X}_2}$ (style loss). The classification output can as well be used in an auxiliary loss such as a label regression.

cost measures the performance of the model within the simulation, by which an optimal parameter configuration is searched until convergence. The dataset is usually split in between training, validation and test data in order to evaluate the generalisation capability of the model. The training set is used for optimisation, the validation set can be used for hyper-parameter search (i.e. empirical tuning or grid-search of the parameters which are not optimised) and the test set is kept unobserved in order to evaluate the model performance on unseen data. Modern deep learning architectures often have a high capacity such that the dimensionality of the model parameter space may be of orders of magnitude approaching that of the dataset, in this case the model is prone to over-fitting (i.e. copying the training data). This behaviour may be observed when the training loss could be extremely decreased while the test loss diverges, a phenomenon which usually means that the model learns irrelevant patterns in the data (e.g. noise) which are detrimental to its performance on unseen test data. Given this common diagnosis, many actions may be taken in order to prevent over-fitting such as early-stopping, regularisation (e.g. weight penalty, dropout), increase of the dataset size (e.g. artificial data augmentation) or adjustment of the model capacity. All these elements belong to

the task simulation within a given dataset and according to a certain optimisation cost, for simplicity and to the extent of this thesis we do not go into details of the many frameworks which may exceed this setting such as reinforcement learning [192], transfer learning [306], one-shot learning [281], neural architecture search [219], meta-learning [116], life-long learning [198] and the ever growing field of artificial intelligence.

The aforementioned practice allows to assess how well a model has learned inside the given task simulation, i.e. according to its training algorithm and the dataset. Although several concerns remain, amongst them is the model robustness outside the dataset and during real-world deployment (i.e. the ability of the simulation to cover unseen situations beyond those of the test set). Another question is the model evaluation, in many cases the end-user objective cannot be strictly evaluated as part of the optimisation and the training objective is only designed as an implicit way for the model to learn the task. For instance, the end goal may not be tractable or it may not be computed efficiently or it may not be differentiable such that error gradients could be back-propagated. While it is hoped that reducing the loss will improve the evaluated model performance, it may not be guaranteed given the gap that exists and there could be cases such that a better performing model with respect to the loss may appear less efficient with respect to the task evaluations. The difficulty of model evaluation notably arises in the field of generative modelling where there are no all-encompassing metrics of the perceptual quality of a model, as opposed to information retrieval tasks which are usually well evaluated by some predictive scores such as accuracy, recall or F-measure. Moreover since the training objective is often model dependent, it prevents direct model comparisons by their losses. These reasons justify the need for evaluation metrics in generative modelling [22] [259] [35] [191] which should ideally be independent from the training algorithms and models, as well as applicable across datasets.

3.3.1 Statistical metrics

Given that most probabilistic generative models are optimising some per-example likelihood (explicitly or implicitly), a natural evaluation is the average log-likelihood of the model on the test set, if tractable, or an approximation based on Kernel Density

Estimation (KDE):

$$\begin{aligned} &\frac{1}{N}\sum_{i=1}^{N}\log(p_\theta(\mathbf{x}_i)) \quad \text{(average log-likelihood)} \\ &p(\mathbf{x}) \approx \frac{1}{N}\sum_{i=1}^{N} K(\mathbf{x}-\mathbf{x}_i) \quad \text{(KDE)} \end{aligned} \tag{56}$$

with kernel function K (e.g. Gaussian). According to the estimated probability functions of the dataset and the model, other measures may be used such as the Kullback-Leibler divergence $\mathcal{D}_{KL}(p_\theta(\mathbf{x})||p_\mathcal{X})$ or the Jensen-Shannon divergence. However, for high-dimensional data these evaluations are not robust such that models with high log-likelihood can produce samples of poor quality [260]. In a similar fashion as the KDE, the Maximum Mean Discrepancy can be used to approximate the distance between the distribution of samples from the dataset and from the model, as well as other methods for two-sample test [158] and coverage metrics [262].

Since blind statistical metrics may not assess well the generative quality of a model, more specific evaluations have been introduced with an emphasis on both the individual sample accuracy as well as the overall diversity, two performances which are often traded-off [1]. Sample diversity can be evaluated via the Number of Statistically-Different Bins (NDB [221]) which states that for a given clustering of the dataset with a bin indicator function $I_B(\mathbf{x}) = 1$ for $\mathbf{x} \in B$, if the model distribution is the same as the data distribution then the number of samples that fall into a given bin should be the same, i.e. $\frac{1}{N}\sum_{i=1}^{N} I_B(\mathbf{x}_i \sim p_\mathcal{X}) \approx \frac{1}{N}\sum_{i=1}^{N} I_B(\mathbf{x}_i \sim p_\theta(\mathbf{x}))$. NDB is measured as the number of bins where the number of data examples is significantly different from the number of generated examples, i.e. a missing mode. The Inception Score (IS [227]) has become a standard generative modelling metric which assesses both accuracy and diversity using a large-scale pretrained classifier (originally Inception Net [256] for images, also extended to audio in [59] [69]). It states that the conditional classifier distribution $p(\mathbf{y}|\mathbf{x})$ over individual samples should have a low entropy (i.e. accuracy as strong confidence at being associated with one of the semantic classes) while the marginal distribution $p(\mathbf{y}) = \sum_{\mathbf{x}\sim p_\theta(\mathbf{x})} p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$ over all generated samples should have a high-entropy (i.e. diversity as uniform distribution over all possible semantic

classes). The IS can be evaluated via the following Kullback-Leibler Divergence:

$$\text{IS} = \exp(\mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x})}\mathcal{D}_{KL}(p(\mathbf{y}|\mathbf{x})||p(\mathbf{y}))). \tag{57}$$

The Fréchet Inception Distance (FID [110]) and its music domain counterpart Fréchet Audio Distance (FAD [139]) use feature statistics computed in the perceptual embedding of a pretrained classifier, akin to the perceptual losses. The means $\mu_{\text{data}}, \mu_{\text{model}}$ and covariances $\Sigma_{\text{data}}, \Sigma_{\text{model}}$ of multivariate Gaussian distributions fitted on the hidden activations of a given pretrained classification layer are compared by the Wasserstein-2 distance:

$$\text{FID} = ||\mu_{\text{data}} - \mu_{\text{model}}||_2^2 + tr(\Sigma_{\text{data}} + \Sigma_{\text{model}} - 2\sqrt{\Sigma_{\text{data}}\Sigma_{\text{model}}}). \tag{58}$$

3.3.2 Data metrics

Individual sample reconstruction errors can be computed to assess a generative model accuracy on test data, while preventing from evaluating the quality of random samples it complements statistical metrics with signal-related distances. Such evaluations are often data-dependent in order to quantitatively assess specific properties related to the perception of images [239], audio [31] and other domains. To the extent of audio quality assessment, there exist specific evaluations for speech [117] such as PESQ [223] and ViSQOL [112] or more generic distances either computed in the waveform domain or on spectrograms. A distance between the reference waveform $\mathbf{x} = \{x_1, \ldots, x_L\}$ and the model reconstruction $\hat{\mathbf{x}}$ can be computed with different measures such as the Signal to Noise Ratio (SNR), the cosine distance (cosdist) or the Czenakowski Distance (CZD):

$$\begin{aligned} \text{SNR}_{\text{dB}}(\mathbf{x}, \hat{\mathbf{x}}) &= 10\log_{10}\left(\frac{\sum_{i=1}^{L}|x_i|}{\sum_{i=1}^{L}|x_i - \hat{x}_i|}\right) \\ \text{cosdist}(\mathbf{x}, \hat{\mathbf{x}}) &= \frac{\mathbf{x}\odot\hat{\mathbf{x}}}{||\mathbf{x}||\odot||\hat{\mathbf{x}}||} \\ \text{CZD}(\mathbf{x}, \hat{\mathbf{x}}) &= 1 - \frac{2\sum_{i=1}^{L}\min(x_i, \hat{x}_i)}{\sum_{i=1}^{L} x_i + \hat{x}_i}. \end{aligned} \tag{59}$$

Waveform-domain distances are sensitive to phase shifts, for this reason they are usually computed in a segmental fashion (i.e. averaged over consecutive windows rather than the whole signal) or replaced by short-term frequency domain distances. Regardless

of whether the model is trained for waveform or spectrogram generation, a spectral distance can be evaluated by measures such as the Spectral Convergence (SC, between magnitude spectrograms $|\mathbf{X}|$), the Log Spectral Distance (LSD), the Itakura-Saito Distance (ISD), the Cepstral Distance (e.g. RMSE between MFCCs) or the Bark Spectral Distortion (e.g. MSE between Bark spectra [307]):

$$\begin{aligned} \mathrm{SC}(\mathbf{X}, \hat{\mathbf{X}}) &= \frac{|| \, |\mathbf{X}| - |\hat{\mathbf{X}}| \, ||_F}{|| \, |\mathbf{X}| \, ||_F} \\ \mathrm{LSD}(\mathbf{X}, \hat{\mathbf{X}}) &= \sqrt{\sum \left(10 \log_{10} \left(\frac{|\mathbf{X}|^2}{|\hat{\mathbf{X}}|^2} \right) \right)^2} \\ \mathrm{ISD}(\mathbf{X}, \hat{\mathbf{X}}) &= \sum \left(\frac{|\mathbf{X}|}{|\hat{\mathbf{X}}|} - \log \left(\frac{|\mathbf{X}|}{|\hat{\mathbf{X}}|} \right) - 1 \right). \end{aligned} \tag{60}$$

3.3.3 Control and task-specific metrics

By measuring the diversity and quality of samples produced by a generative model, one can quantitatively evaluate the extent to which the model has succeeded in fitting the target data distribution. Signal-related metrics emphasise the reconstruction accuracy of the model in domain-specific features as an automatic assessment of the perceptual quality. In the audio domain, the perceived quality is crucial for telecommunications (e.g. intelligibility, fidelity of the individual voice) as well as for music applications which cannot compromise the realisticness of the sound as well as its aesthetics. Nonetheless, the usability of a generative model directly depends on the controls it offers and the interpretability of its representation with respect to the end-user task. A taxonomy of classical synthesis techniques and corresponding evaluation criteria for control parameters have been proposed in [130] [261], of which we summarise the main qualities:

- ▷ Intuitiveness: a well-behaved control parameter should have a perceived effect which is proportional to the parameter variation and this response should be homogeneous and accurate across the parameter range.
- ▷ Interpretability: the perceived effect should be predictable such that there exist a meaningful relationship between the expected outcomes and the control parameters. This quality can be promoted by a sparse control stream (fewer observed

parameters or macro-controls) or an underlying physical model to which parameters can be related. This quality can as well benefit from analysis tools which describe the relationship between parameters and data observations or metadata.

- ▷ Diversity: which amount of variations and classes can be represented within the control parameter ranges.
- ▷ Robustness: a given perceptual class should be consistently identified across its corresponding parameter range and the overall parameter range should yields outputs which remain consistent with the expected classes.

We can observe that statistical and data metrics assess part of these criteria, mainly the diversity and robustness. On the other hand, the intuitiveness and interpretability are little to not covered although they are crucial elements of the end-user interaction. Conditional density estimation allows to define a specific information structure and semi-supervision lets enforcing the use of specific attributes as part of the generation process. It is thus relevant to evaluate how accurate is the semi-supervised generation with respect to the pre-defined attribute targets. A task-specific classifier can be trained as a reference for automatic evaluation of a generative model, which differs from large-scale generic classifiers used as embeddings for statistical metrics. For instance, a pitch-conditional GAN is trained in [69] along with a reference pitch classifier in order to measure the accuracy and entropy of synthesised outputs with respect to their pitch targets. In a similar way, one could train an instrument classifier in order to assess the accuracy of domain translation in between timbres or to measure the robustness of timbre while varying pitch conditions. These approaches could be as well extended to continuous conditioning parameters given a reference regression model, for instance replacing pitch classification with fundamental frequency estimation. Such attributes which can be labeled are inherently interpretable, e.g. they have a physical sense such as pitch, thus if learned accurately they are as well intuitive to the user. Other properties such as continuous timbre variations do not have an explicit measure, which is a common problematic to evaluate many kinds of semantic interpolations. Inference models benefit from the possibility to analyse data and visualise their relationships in the continuous embedding, this gives insights in the latent structure from which one can continuously generate. In the conditional setting, it is reasonable to consider the latent representation as an embedding of the data variations which are not specified by the conditioning attributes [156], for instance by conditioning on pitch and loudness we

can expect that the embedded factor of variations mainly account for timbre. Yet, the unsupervised latent variables are often hard to interpret and hardly intuitive as control parameters. Unfortunately, metrics for interpolation [15] and disentanglement [111] are often only applicable to simple simulated datasets or limited subsets of properties of real-world data. One approach to measuring the representation quality is to perform post-classification on the learned features with a low-capacity model. Its underlying idea is that if an unsupervised representation is efficiently structured, it can easily be transferred to train a shallow classifier. For instance an instrument classifier with a single layer over latents, which would be ineffective if directly applied in the audio domain. Since the latent space dimensionality of a generative model is often impractical to visualise (e.g. more than 3 dimensions), interactions can as well be enhanced by using an invertible dimensionality reduction technique such as PCA [83] although it discards parts of the data variations (i.e. dimensions of lower variance).

3.3.4 Human rating

Significant research is invested in developing automatic evaluation metrics although there remain a gap between computational assessment and the human judgement which cannot ultimately be replaced. Particularly in the field of generative modelling and neural audio synthesis, human rating often remains necessary in order to measure the perceived realisticness (e.g. Turing test) and subjective user preferences. To this extent, crowd-sourcing platforms such as Amazon Mechanical Turk have been commonly used to carry listening tests and collect Mean Opinion Scores (MOS [252]) from randomised A/B tests [151]. The rating is often based on a scale of 5 points and averaged over many listeners and randomised listening tests which yield a MOS. It is also a common practice to evaluate the perception of real data alone (e.g. ground-truth audio) in order to report an upper bound of the experiment as the perceived quality of real data may be rated lower than 5. Human assessment gives valuable qualitative insights on the generative performance, however it is time-consuming, expensive and less reproducible than automatic evaluations. Another kind of human-in-the-loop evaluation which is less considered in the academic research, although probably highly valued in the industry, is expert user feedback for instance on the usability and controllability of a system. While MOS often relies on the innate human ability to listen and judge, the feedback on usability requires trained practitioners and even longer testing phases.

3.3.5 Efficiency

Deep learning has thrived on the increased computational resources enabled by specific chips such as Graphics Processing Units (GPU) and Tensor Processing Units (TPU). These processors allow accelerated parallel computing which is a backbone of all modern machine learning frameworks, nonetheless the overall training costs in deep learning research have steadily increased as the capacity of models can be augmented to unprecedented sizes in order to push the state-of-the-art performance in many domains and tasks [230]. This raises several issues and a growing concern in the academic and industrial community about the energy footprint of neural network training (e.g. grid-search optimisation) and life-long deployment [253], as well as the inclusivity of the research for institutions with unequal budgets [107]. Training extremely large networks [241] on gigantic datasets can yield increased performances, yet the evaluation dramatically lacks taking into consideration efficiency, i.e. the trade-off between exponential increase in computational costs and the magnitude of the score improvement [119] (e.g. only a few percents of accuracy). When it comes to measuring the efficiency of a model, several factors could be taken into consideration such as its training time and inference speed with respect to the computing resources engaged, the amount of data required to train, the diversity of tasks and data it can process, its transferability to other downstream tasks [213]. Pruning methods for optimising a given model to achieve resource efficient inference [189] have a long-standing history in machine learning research [218]. The effectiveness of pruning may be understood by the fact that the best generalization will be achieved by the smallest system that will fit the data, as opposed to overfitting. Nonetheless, there is no definitive rule on how to set this optimum size and neural networks are often over-parametrised in order to allow a sufficient flexibility (i.e. large search space) and successful optimisation. As a result, only a fraction of parameters out of the random initialisation effectively contribute to the performance of the trained model. Pruning methods usually rely on some ranking criteria of the importance of parameters in order to mask the least effective ones and fine-tune the sparsified model to maintain a target accuracy or even outperform this accuracy after model compression [305]. Amongst the recent developments of model compression techniques [20], unprecedented gains in performance and sparsification have been achieved based on the lottery ticket hypothesis [80] [93]. Instead of the usual pruning by fine-tuning, the lottery ticket hypothesis relies on rewinding a trained model to its randomly initialised state, masking the weights identified as least relevant and re-training. After

several iterations, the amount of active parameters can be drastically reduced and the performance increased by significant margins. The pruning results achieved in a broad array of tasks and frameworks highlight the need for taking into consideration efficiency as a central evaluation of deep learning, since they demonstrate an obvious trend to over-parametrisation which is detrimental to generalization and induces a significant environmental footprint [5].

In the field of audio and music, efficiency of deep learning is an essential fact to keep in mind since many applications require low-latency or real-time capable processing on constrained devices (e.g. mobile phones, laptops, embedded devices). Classical DSP solutions are very efficient compared to deep learning solutions, this allows regular laptops to flawlessly run many softwares in real-time and in parallel for instance a Digital Audio Workstation with multiple VSTs. While it is common place in any music production workflow, it is up to date unrealistic to integrate deep learning models in such setting. Since local integration is often impossible, modern technologies powered with deep-learning usually rely on cloud computing which introduces an inherent latency, issues with stability, continuous costs as well as concerns on security and privacy since one constantly needs to be connected and exchanging data.

4 Related Works in Neural Audio Synthesis

As exposed in section 2.4, data-driven approaches to audio and music processing have been first developed for information retrieval in a supervised setting. Over the last decade, significant advances have been done in the unsupervised generative setting and major breakthroughs could be first witnessed in the field of computer graphics and neural image rendering [258]. Generative modelling has gradually extended to the audio domain and is currently an active field of research of its own. Neural audio synthesis was primarily driven by the speech community, more recently it has been spreading into the music domain which is the focus of this thesis. To this extent, we present the related works structured as follow:

- ▷ Unconditional audio synthesis: we review reference works which are general to the domain of neural audio synthesis, as such they have been both applied to speech and music synthesis, as well as extended to conditional generation and specific tasks.
- ▷ Spectrogram inversion: from the perspective of Figure 6, neural audio synthesis is a multi-scale problem and intermediate short-term representations such as magnitude spectrograms are often interleaved between the global context and local synthesis. This is observed both in the speech domain (e.g. Text-To-Speech) and in music generation (e.g. score to audio), which can be performed in two steps by first generating spectrogram and then inverting to the audio waveform. This approach breaks-down the challenge of audio modelling on long temporal scales, thus spectrogram inversion models are common elements of a neural audio synthesis pipeline [240].
- ▷ Implicit timbre models: we introduce conditional generative models specifically applied to music, as opposed to unspecific inversion models conditioned on spectrogram. Timbre is a central element in the task of neural audio synthesis, in the first place we review reference works done in implicit timbre rendering. These models do not have explicit parameters of timbre, instead they are mostly conditioned by example (i.e. domain translation) or by disentangling other acoustic features from the model representation (e.g. fundamental frequency and loudness). When framed in the analysis/synthesis approach, these models often rely on hand-crafted feature extraction.

- ▷ Learning representations of Timbre: since continuous timbre variations cannot be explicitly labelled, we review generative models that learn unsupervised and semi-supervised representations of timbre. One major challenge is to find interpretable and disentangled representations in order to allow intuitive timbre control from the learned representation. When framed in the analysis/synthesis approach, these models often rely on learned feature extraction (e.g. auto-encoder).
- ▷ Score to audio: models which are conditioned on high-level musical context and translate a whole composition from the symbolic domain to the acoustic domain. Although we focus on the synthesis process, we as well discuss bijective systems which perform both symbolic inference and acoustic generation.
- ▷ Perceptual audio embedding for generative modelling: we detail some models relevant to section 3.2.5 which are specifically trained in the audio domain.

4.1 Unconditional Audio Synthesis

The primary aim of neural audio synthesis is to learn generative processes that match the distribution of audio observations in order to consistently synthesise new audio data. In that sense unconditional synthesis is commonly associated with probabilistic models that fit a data density solely based on the audio observations, without any conditioning context, and in turn can generate new data by sampling the learned density.

Auto-regressive models. Audio information is by nature a temporal process and its lowest-level representation, the waveform $\mathbf{x} = \{x_1, \ldots, x_L\}$, a uni-dimensional causal time series. Auto-regressive density estimation (Figure 20) is thus a relevant approach, yet it faces the challenge of the high sampling rate of audio which produces time series of large dimensionality. Given the auto-regressive model formulation $p_\theta(x_i|x_{<i}) = p(x_i|G_\theta(x_{i-1}, \ldots, x_{i-T}))$ with G_θ a neural network that estimates the next time step based on its receptive field over the past T time steps, we can observe that inferring from a limited context of 250 milliseconds at 44.1kHz would already require modelling dependencies across 11025 dimensions. Due to this computational complexity, models are often developed at lower sampling rates such as 16kHz although it prevents from rendering the higher part of the audible spectrum. WaveNet [268] has pioneered raw waveform generation using an auto-regressive architecture based on stacking dilated

convolutions which allow an exponential increase of the receptive field for a given model size. It achieves a receptive field of about 250 milliseconds which is sufficient for synthesising a locally consistent audio signal and uses μ-law quantised waveform amplitudes (8-bit) in order to optimise the conditional prediction as a compressed categorical distribution of 256 bins. An alternative approach to increase the learned context is proposed in the Sample-RNN [185] architecture which uses parallel RNN tiers that operate at fractions of the raw sampling rate. The idea of partitioning a RNN network into modules operating at different time resolutions was introduced in the Clockwork-RNN [153] in order to facilitate memorising longer-term dependencies, although initially limited to waveform segments of 7 milliseconds. The Sample-RNN output layer predicts μ-law quantised waveform samples while upper-level RNNs process down-sampled representations and aggregate memories over increasing temporal contexts. Recurrent units do not have an explicit receptive field yet in practice they only learn to memorise a limited context, the Sample-RNN multi-scale architecture is able to model audio dependencies consistent in the order of a second.

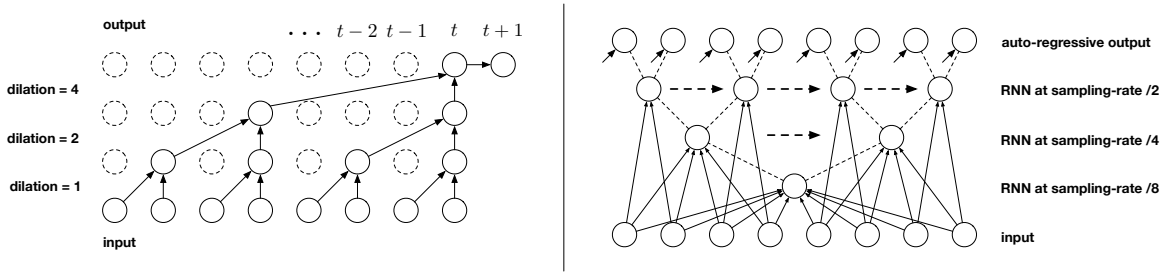

Figure 20: Left: Stacked convolutions as in WaveNet with a receptive field 2^n growing exponentially with the number n of layers. Right: A pyramidal RNN architecture as in Sample-RNN with tiers operating at fractions of the sampling rate in order to aggregate memory over longer contexts to condition the auto-regressive output prediction. Each tier (a row) is an individual RNN and dashed connections are the conditioning from one tier to another. Horizontal arrows show the strided translation along time of each RNN tier.

Auto-regressive audio modelling raises several issues amongst which are the high model complexity required for inferring from limited contexts, inducing long training times and needs for large datasets, the slow generation since sampling is done one time step at a time and the optimisation strategy. Using μ-law quantisation simplifies the optimisation, yet auto-regressive models often need to be trained with teacher-forcing (i.e. using ground-truth past samples) which causes an exposure bias at inference time (accumulating errors from the prediction into the past context). By design the auto-regressive training objective is sensitive to phase shifts which is one cause of the

optimisation difficulty, according to it two waveforms with a phase offset will be strongly penalised although they are equally perceived by the human audition. Auto-regressive modelling in the magnitude spectrogram domain is proposed in [271] to address learning longer temporal dependencies with a phase invariant representation. Since the short-term time axis is down-sampled, this model learns consistent temporal structures of the order of several seconds. However it prevents from high audio fidelity and end-to-end learning as it only models lossy features (discarded phase, Mel scaled frequency) which require a subsequent inversion model. Since the adoption of WaveNet as a reference baseline in neural audio synthesis, we can observe that a significant amount of research has been pursued into improving audio modelling directly in the waveform domain [21]. In order to increase the temporal receptive field of auto-regressive waveform models and generate audio with consistent longer temporal structures, [54] proposes to learn multiple WaveNet auto-encoders which operate at different scales. By pooling the output of WaveNet encoders, the model aggregates a downsampled context which provides a hierarchy of decoders with signals corresponding to features of increasing temporal scales. In order to allow faster than real-time sampling, the Parallel-WaveNet [269] and ClariNet [208] introduce a two-stage training which relies on using a pretrained WaveNet as teacher to train an Inverse Auto-regressive Flow (IAF [147]) student model. The IAF model allows fast parallel sampling but is slow to train due to the sequential likelihood estimation. Its training is made efficient by a method called probability density distillation, which optimises the student sample generation under the distribution learned by the teacher. Once the IAF student has matched its teacher distribution, it can be used as a feed-forward replacement of the pretrained WaveNet without loss in audio quality, although the overall process requires twice the amount of training and the teacher-student matching is prone to instability. Alternatively, a highly optimised single layer RNN is proposed in [135] for waveform modelling with a dual softmax output over 16-bit audio at 24kHz. The proposed model is pruned and fine-tuned, this enables a network sparsification by 96% without loss of audio quality and efficient inference on mobile phone CPU.

Adversarial Audio Synthesis. Oppositely to auto-regressive models which are biased towards learning local dependencies with an iterative sampling scheme, Generative Adversarial Networks allow fast parallel sampling from a latent prior and global evaluation by an adversarial discriminator. The GAN framework has largely contributed to

advances in neural image rendering, however it faces challenges in the audio domain because of the sequential structure of the data (as opposed to images with a fixed spatial size) and the need for locally coherent phase generation to avoid strong auditory artefacts. In [59], a DCGAN image baseline is unrolled on the time axis by flattening both kernels and strides and allows producing consistent audio waveforms such as speech fragments or musical samples (e.g. drum hits). The up-sampling artefacts caused by transposed convolutions [193] are mitigated by nearest-neighbour up-sampling in the generator and phase shuffle in the discriminator to prevent from learning a trivial policy based on phase patterns. Alternatively to modelling raw waveform, adversarial audio synthesis has been applied to time-frequency representations designed for efficient inversion in [69] [177]. Since the raw phase information is highly unstructured, which disrupts the neural networks efficiency for pattern recognition, alternative representations are paired to the magnitude spectrogram by taking either phase time derivatives (e.g. instantaneous frequency) or phase frequency derivatives (e.g. group delay). These models benefit from efficient learning in the time-frequency domain and approximate inversion to waveform is reported to achieve high audio quality and fast rendering. However, the aforementioned models are bound to generate fixed length audio (in the order of a second) from a unique global latent prior, which is a strong limitation that is tackled in [163]. The proposed model generates magnitude spectrograms of arbitrary length in a coarse to fine manner from series of random noise vectors sampled from the prior. However it relies on a subsequent inversion model to synthesise waveform from the lossy Mel-spectrogram output, as opposed to the prior works which target end-to-end learning.

4.2 Spectrogram Inversion

Spectrograms are rather generic representations that are applicable to many types of audio domains, such as speech, environmental noises and musical sounds. Because of the down-sampled time frame axis, generation using an intermediate time-frequency representation can help breaking down the complexity of neural audio synthesis into two steps. The understanding of the global and long-term context (e.g. language, composition) can belong to a first model generating acoustic features in the spectrogram domain. This step is usually performed without modelling the unstructured phase information [282], thus it requires a second inversion step to waveform which can be

performed with the standard Griffin-Lim Algorithm (GLA [100]). Although convenient, this approach prevents from end-to-end learning which leads to error accumulation of both the spectrogram prediction and the waveform inversion. Using GLA notably introduces auditory artefacts from the approximate phase estimation, as well as latency as many iterations are required. Based on that, the desired properties of the spectrogram inversion model are the audio quality (compounding as little as possible errors) and the synthesis speed (adding the least latency after the spectrogram generation).

WaveNet-like vocoders. Because WaveNet has proven an unprecedented quality at generating locally consistent waveform, it was subsequently applied to spectrogram inversion by providing it a global context in the form of spectrogram conditioning [240]. In this sense, the conditioning is provided as a guide but does not correspond to the control definition in Section 3.3.3. Interactions are usually learned in the spectrogram generation model which conditions the inversion model with a dense acoustic specification of the audio target which should be rendered with the highest possible fidelity (no semantic alteration). Many research investigations have been dedicated into WaveNet inspired vocoders which alleviate the limitation of slow iterative sampling while maintaining its high audio fidelity. As observed in Parallel-WaveNet, an IAF student can be trained to this extent and subsequent experiments have developed generative flows for fast feed-forward waveform synthesis without a two-stage training. In order to parallelize both training and generation, these models have mostly revolved around the affine coupling transformation [57] which allows fast computation of the jacobians and arbitrary operations over the conditioning signal (Figure 21). In WaveGlow [210], the audio counterpart of [144], a uniform noise prior distribution is mapped to waveform given spectrogram conditioning by stacking several invertible transforms and optimisation is directly performed on the exact likelihood. The signal is arranged in contiguous channels by a squeeze operation so that each time step covers an increased duration, the spectrogram is aligned along this axis and processed by dilated temporal convolutions which provide parameters of the channel-wise affine transforms. Several models have built upon this framework and provide modelling improvements in order to speed-up the training and inference [143] [209] [297] [140].

Alternative vocoder formulations. Other architectures and training schemes have been proposed besides the ongoing effort spent into refining variations of the WaveNet

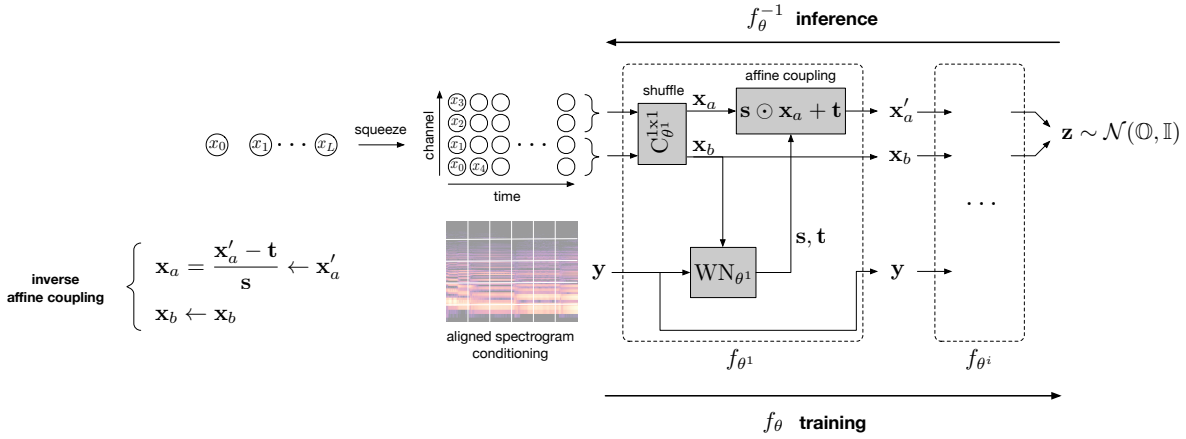

Figure 21: Schematic view of a generative flow conditioned on magnitude spectrogram, such as in WaveGlow. The audio waveform is squeezed into channels, aligned with the acoustic conditioning and processed by a stack of invertible affine coupling transforms. Half of the squeeze dimension is fed with the conditioning into a WaveNet-like module that predicts biasing and scaling coefficient $\mathbf{s}, \mathbf{t}$ *to transform the other half input. The invertible* $1x1$ *convolution allows mixing the squeeze channels at each flow step.*

vocoder [96]. The FFTnet [131] uses a tree structure such that each layer splits its input into even and odd steps that are transformed and summed, akin to sinusoidal and cosinusoidal components of a DFT. By stacking n of these transforms it aggregates a context of size 2^n, two such networks are combined to process both past audio samples and aligned acoustic features by summing their hidden features. The output node of this tree is used for conditional auto-regressive prediction and it is reported to achieve a competitive audio quality with a lighter network. Apart from auto-regressive and likelihood based neural vocoders, fully-convolutional feed-forward models have been proposed in order to synthesise several orders of magnitude faster. By design, these models are parallel for both training and sampling thus potentially very efficient. The Multi-head CNN (MCNN [9]) takes advantage of the additive property of sound by using several parallel networks which focus on up-sampling different frequency bands of the input spectrogram and sum up into the output audio. Its optimisation is based on hand-tuned spectral losses for both magnitude and complex components which guide the reconstruction of specific perceptual audio qualities.

The GAN framework has also been applied to spectrogram inversion such as in MelGAN [154] which is able to produce variable-length audio with an unprecedentedly fast and efficient architecture. To that extent, the authors introduce specific architectural choices for both the generator convolution parameters (kernel, stride, dilation sizes)

which alleviate up-sampling artefacts and for the discriminator. It is composed of modules operating on several down-sampled audio resolutions in order to discriminate over different frequency ranges. This discriminator is used both for matching the expected dataset density and for a deep feature loss which ensures that the generated audio is accurately conditioned. This idea of multiple discriminators is also developed in [19] which proposes an ensemble of random window discriminators to capture both different resolutions and randomised sub-contexts as a mean of data augmentation. Other GAN developments combine a multiple discriminator approach with a multi-band generator architecture [292], add nested discriminators at the intermediate layer outputs of a multi-scale generator [293] or pair the GAN criterion with a multi-resolution spectrogram reconstruction loss [291].

Alternatively, some neural vocoders more closely follow the original Griffin-Lim algorithm and implement its successive iterations of phase estimation as the chain of transformations within a feed-forward neural network. The GAN framework is used in [195] to predict the correct complex spectrogram from a magnitude and zero-initial phase, which in turn can be directly inverted with the inverse STFT. In [179] the GLA iterations are learned with a stack of neural networks trained in a denoising fashion by reconstruction of the clean signals from their complex magnitude spectrograms altered with random noise perturbations. Whereas most generative models estimate a target data density or regress some reconstruction errors, generative models based on score-matching directly learn the gradients of the target data density. This approach is applied to conditional audio synthesis in [34] [152] which iteratively convert a random noise into waveform by gradient ascent in the data space, i.e. iteratively maximising the waveform likelihood for the given acoustic context. At each denoising step the network is provided with the spectrogram conditioning and the current noisy signal and then directly estimates the gradient direction to refine the signal estimation. These models are non auto-regressive and offer an explicit trade-off between synthesis speed and quality by setting the number of denoising iterations (akin to GLA). It is shown to achieve high audio quality, competitive to state-of-the-art auto-regressive solutions, with compact architectures that can be parallelized to arbitrary signal durations thus allowing efficient sampling.

Learning invertible time-frequency representations. Usual time-frequency transformations such as the STFT and its inverse are convolution operations with a set of

filters which define the frequency basis (e.g. in Equation 2 the Fourier basis with real and imaginary parts computed with cosinusoidal and sinusoidal filters). Accordingly, these transformations can be efficiently computed with 1-dimensional convolutional neural networks and these filters may be adaptively learned as part of an analysis pipeline for MIR [36] or speech recognition [217]. Moreover, research in source separation has seen recent breakthroughs [170] [197] by learning invertible time-frequency representations as part of the analysis-separation-synthesis pipeline commonly used for this task. Earlier works in the field have used fixed transformations for the analysis and synthesis (e.g. STFT/iSTFT in [128]) and these results have been largely improved by performing the separation (e.g. masking) in the analysis/synthesis representation learned with neural networks. Although we have not seen yet any experiments applied to generative modelling, one could apply this technique by learning an invertible time-frequency representation instead of a neural vocoder which approximates the inverse transformation. This could allow end-to-end learning, as in source separation where the representation is learned along with the masking, by modelling an upper-level generative task in the time-frequency representation learned with a lower-level analysis-synthesis network.

4.3 Implicit Timbre Models

The aforementioned models for neural audio synthesis are mainly generic and make little to no assumptions on the types of sounds to be represented. To the extent of this thesis we focus on musical audio and the predominant acoustic features to be modelled are the fundamental frequency, which translates into pitch and harmony, the loudness, which translates into velocity and dynamics, and the timbre which carries the perceptual identity of the instrument. There surely exist some inter-dependencies across these features, for instance the pitch is bounded by the instrument tessitura and playing styles induce variations both in timbre and in fundamental frequency and loudness envelopes (e.g. vibrato is a pitch modulation, tremolo is a loudness modulation). Nonetheless, this gives us a standard categorisation of acoustic features and it conveniently aligns with our understanding of the music generation pipeline as pitches and dynamics are composition elements (as well as rhythm is) which specify a content from which the synthesis model should render all the acoustic details which belong to the timbre domain. From this perspective the fundamental frequency, if the considered sources are pitched instruments, and loudness are inherent specifications to the task of neural audio syn-

thesis for music and the models may be categorised by their representation of timbre. These representations notably vary according to either treating timbre as a categorical feature (an instrument class), an implicit feature (e.g. the remainder of pitch and loudness) or a continuous feature (expressive representation), as well as by the number of timbre domains that can be processed (e.g. single or multiple instruments).

Neural style transfer for audio. The results achieved in the domain of neural image stylisation [88] have motivated subsequent experiments to apply this technique to audio generation, for instance timbre transfer when providing content and style samples belonging to different instruments. Neural stylisation is directly applied to magnitude spectrograms in [274] by computing the content and style losses in the embedding of a pre-trained audio classifier in order to match a randomly initialised spectrogram to the target feature statistics. This naive application raises two main questions, the choice of representation which treats spectrograms as images and the lossy generation since the magnitude spectrograms are inverted with GLA. Different representations and corresponding pre-trained embeddings are investigated in [187] which opens the possibility of waveform domain audio style transfer. Further works [101] show that the randomly initialised input can be replaced with the content sample and optimisation may only be performed on target style features, a process which resembles more to domain translation. These applications of neural style transfer for audio have opened some creative approaches for implicit timbre synthesis driven-by-example and were refined for texture synthesis in [6], yet they do not account for many of the intrinsic differences between the audio and visual domains. Audio representations are temporal and do not satisfy the same statistical properties as images [290], moreover the semantic of music is multimodal and the optimisation derived from Gram matrix statistics in a single generic embedding is unlikely to disentangle what is implicitly provided as style and content within the acoustic mixture. To this extent, generative modelling approaches to domain translation may learn more adapted features which globally belong to an instrument or a music style class and are expected to disentangle arbitrary contents (domain-invariant) and styles (domain-specific) from the acoustic mixture, for instance using unpaired data and adversarial learning. To the extent of this thesis, we focus on timbre style transfer and attributes pertaining to the acoustic domain. It should be noted that this process is only a subset of music styles which also encompass composition style transfer and performance style transfer [184].

Domain translation for timbre. While the aforementioned approach to audio style transfer iteratively performs a cross-synthesis in between two chosen samples, domain translation aims at modelling the transformations between datasets and can be trained without supervision (e.g. paired data). In this setting, the model should learn features for both the realisticness with respect to the target (e.g. the domain style) and the uniqueness of the mapping which preserves the underlying content of the given input. The TimbreTron model [120] applies a CycleGAN [304] on a CQT audio representation for bijective timbre transfer between individual instrument notes of fixed-length. This image-like representation is chosen because of the pitch equivariance in the frequency axis, accordingly translations in the 2-dimensional spectrogram are perceptually consistent (linear temporal lag or linear pitch transposition), which approximates the invariance property of the spatial dimensions of natural images. In addition to the usual GAN losses that ensure that samples generated in a given domain are realistic, the cycle-consistency losses ensure the uniqueness of the mapping by pushing the generators to invert each other. Waveform synthesis from spectrogram is done with a pretrained WaveNet conditioned on CQT predictions. Music conversion in between similar sub-genres of electronic dance music is proposed with CycleGAN in [272]. The model is applied to magnitude spectrograms of 4-bar long music excerpts and does not require specifying instruments or isolating tracks. In this setting the style differences are mainly attributed to textures (e.g. harshness of synthesizers, intensity of drums) and the conversion does not modify the music structure, akin to an audio effect. Direct iSTFT spectrogram inversion can be performed by using the source phase information because audio texture differences are little sensitive to phase. A variable-length audio style transfer model is proposed in [199] by adapting the TraVeLGAN [4] to spectrograms. Pairs of contiguous spectrogram slices are split before one-sided domain translation and concatenated back at the input of the discriminator which assesses the realisticness of the generated result with respect to the target domain as well as to the frame continuity. Because the generator output should be free from artefacts and discontinuities at the concatenation edges, it is able to process variable length spectrograms. In order to learn a style transformation which preserves content information (e.g. music structure, speech intelligibility), an additional Siamese network is trained in cooperation with the generator to preserve vector arithmetic in the cross-domain embedding. This approach was proven efficient for image translations between more heterogeneous domains, at which cycle-consistency fails, and its application to audio allows conversion between more dissimilar genres such as pop and classical music domains.

Multi-domain timbre transfer and variable-length waveform modelling are performed in [190] by training a universal encoder and domain specific WaveNet-based decoders. The domain invariance of the encoder output is enforced by adversarial training against a latent classifier of the source domains, akin to [156]. This regularisation lets multiple decoders share the latent representation in order to generate audio in their respective domains and the experiments show that the pre-trained encoder can be transferred to train new decoders on unseen domains, for instance adding a new target instrument. However the model computation and data requirements are prohibitive since it relies on as many WaveNet models as there are target domains. Another limitation shared by all the aforementioned models is that their mappings do not allow finer controls than selection of a target domain, nor they allow generating multiple candidate conversions for a single source example. This challenge is tackled in [167] which learns a bilateral multi-modal timbre transfer by adapting the MUNIT framework [122] to audio spectrograms. This model combines two auto-encoders with disentangled style and content latent features that are trained with domain specific discriminators. Besides the intra-domain reconstruction, the learning of domain invariant content and domain specific style features relies on the cross-domain adversarial loss and cycle-consistency. Given a source content encoding and random target style code, the decoded sample must appear realistic to the target domain discriminator. Moreover the encoding of that generated sample from the target domain should be consistent such that its style matches the provided random style code and its content code matches the one of the original source encoding. A multi-channel spectrogram representation is proposed in order to account for different qualities of timbre and their acoustic correlations, it comprises the Mel spectrogram and three subsequent features: MFCC, spectral difference and spectral envelope. According to this correlated multi-channel representation, the MUNIT losses are complemented with an intrinsic consistency loss which assesses that the generated spectrogram features follow their pre-defined relationships to the Mel spectrogram channel. Audio synthesis is performed only using the Mel spectrogram channel that is mapped to linear scale and inverted with iSTFT using the phase from the source sample. Because the style is randomly sampled from a Gaussian prior when training the conversion, it allows multi-modal outputs by combining any given source content features with different target style codes, including those encoded from target data observations.

Implicit acoustic models. Implicit timbre modelling driven by example such as in domain conversion allows unsupervised learning and sound transformations, yet it does not offer an expressive acoustic control over the generated outputs. Some low-level acoustic properties such as the fundamental frequency [142] and the loudness can be automatically extracted to condition fine-grained neural audio synthesis models which convert acoustic envelopes into audio. These acoustic properties are extracted at a slower frame rate than audio (e.g. control rate of 250Hz), similarly to spectrogram features, but only provide a partial acoustic specification of the target audio as opposed to spectrogram conditioning. For instance, pitched instrument synthesis may be controlled by the f0 and loudness envelopes from which timbre is implicitly generated as the remainder of acoustic features. Thus we refer to this class of models as implicit acoustic models which learn to generate a specific timbre (e.g. an instrument) based on its corresponding fundamental frequency and loudness envelopes. This approach offers direct and independent controls for instance by transposition of the f0 without alteration of the loudness. These models can be applied to timbre transfer using the acoustic bottleneck of the control features, for instance extracting envelopes from one source instrument performance to condition the synthesis of another target instrument learned by the implicit acoustic model.

The WaveRNN architecture is adapted to musical audio synthesis from f0 and loudness envelopes in [104]. This experiment explores different conditioning strategies such as representing the f0 with a categorical pitch embedding and continuous cent deviations, which are fed in separate stacks processing the categorical and continuous feature envelopes. The output of the conditioning network is up-sampled to the audio rate and used as conditional biasing in the auto-regressive waveform prediction with a highly optimised single layer RNN cell. An implicit model for both timbre and articulation is proposed in [186] by hierarchical in-painting at increasing sampling rates (Figure 22). The model uses the input f0 conditioning to synthesise a pure sinusoid tone at 2kHz that is wave-shaped by a convolutional generator which is expected to generate a realistic articulation of the target fundamental frequency. The output of each wave-shaping module is up-sampled by a factor of two before feeding the next level of sampling rate until reaching the target audio rate (e.g. from 2kHz to 16kHz). Because each step increases the Nyquist frequency by two, the model iteratively enriches the spectral distribution of the target timbre by in-painting the upper half added to the spectrum of the previous scale. The coarse-to-fine generation, from the articulation (f0

contour) to the target timbre, is conditioned by the loudness up-sampled to each scale resolution and trained with a spectral reconstruction error, an adversarial discriminator and a perceptual loss computed in the embedding of a pre-trained pitch extractor. The composition of these losses is expected to ensure that the model follows a consistent pitch trajectory while flexibly shaping a realistic articulation and timbre distribution.

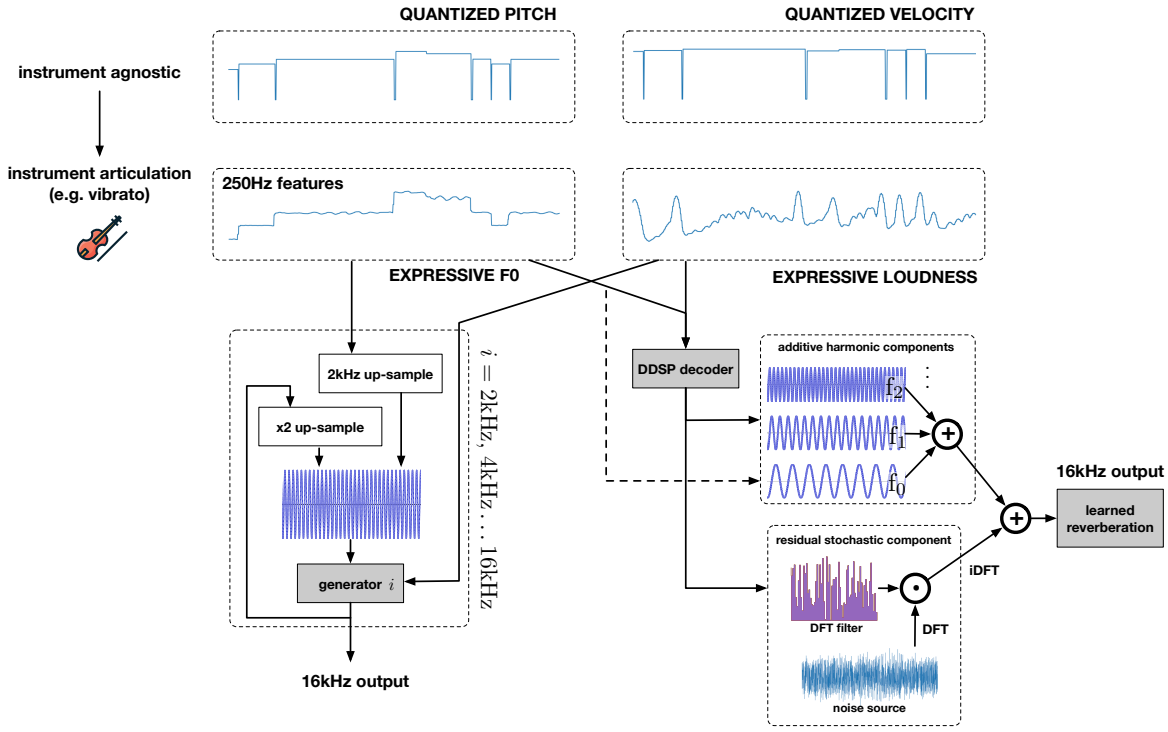

Figure 22: Neural audio synthesis from fine-grained fundamental frequency and loudness envelopes. On the bottom left, the hierarchical in-painting approach with generators that waveshape a pure tone input at several up-sampling stages up to the target audio rate. On the bottom right, the DDSP model for neural audio synthesis parametrised with a harmonic additive synthesizer summed with a subtractive synthesizer (spectral domain noise filtering). The DDSP framework combines interpretable modules and back-propagation in order to train efficient and modular neural networks for audio processing, for instance by chaining the output synthesizers with a learned convolution module. These two models are conditioned with expressive continuous input envelopes at a relatively high control rate (e.g. 250Hz), thus not directly applicable to composition. One further direction of research is to learn control models able to convert the quantised score targets into realistic acoustic features for conditioning the DDSP decoder.

The aforementioned models allow expressive control for fundamental frequency and loudness but their interpretability is limited as WaveRNN relies on a highly optimised RNN cell and the hierarchical in-painting model relies on a hybrid architecture that combines many sub-networks and various losses including adversarial training.

A highly efficient and interpretable approach to neural audio synthesis is introduced with the Differentiable Digital Signal Processing (DDSP [70]) library which combines elements of classical DSP that were specifically implemented for parallel computation and back-propagation in order to be integrated within the deep learning framework. It is observed in both the audio domain and in the image domain [138] [137] that integrating elements of classical rendering within generative modelling can allow learning very efficient, interpretable and controllable architectures. Accordingly, these models directly benefit from domain-specific knowledge and are biased to learn properties of the signal perception. The DDSP decoder implements the SMS decomposition [237] by inferring control parameters for an additive harmonic synthesizer which is summed with a subtractive noise synthesizer to render both deterministic and stochastic audio components (Figure 22). The decoder outputs for amortised synthesis are the harmonic amplitudes of the additive harmonic synthesizer and the spectral amplitudes of the filter applied to the noise spectrum, which are predicted with a rather generic neural network fed with f0 and loudness envelopes. To this extent, it should be noted that the additive synthesizer (following Equation 7 with zero initial phase) is directly fed with the pre-extracted input f0 [142] up-sampled to audio rate, thus it cannot modify the articulation or correct some potential prediction errors. A learned convolutional reverberation module can be trained at the summed synthesizer output in order to disentangle the average recording reverberation from the decoder. This architectural modularity allows direct audio manipulations such as dereverberation by reconstruction without the reverberation module. Due to the specific model design, it achieves high quality waveform generation after a few hours of training on little amount of data (e.g. 20 minutes of audio) and low-latency inference. The DDSP framework is a very promising direction for musical neural audio synthesis and its current limitations raise many questions for future research:

- ▷ Implementing a more flexible sound model within the DDSP framework. The harmonic plus noise decoder output can approximate the ordinario playing style of wind, brass and bowed string instruments but it does not model well transients and percussive components (e.g. violin pizzicato). It cannot accurately model other families of instruments such as pitched percussions (e.g. timpani), plucked/struck strings (e.g. guitar, piano) and inharmonic instruments (e.g. bells). One such attempt has been done for generalising DDSP to speech in [75] and could potentially apply to singing voice synthesis.

- ▷ Handling of polyphonic audio with multiple simultaneous fundamental frequencies or non-pitched audio sources (e.g. drums).
- ▷ Modification of the articulation as part of the timbre transfer. As the input f0 is directly fed into the additive synthesizer output, the model does not have the flexibility to modify the articulation and can fail at producing high-quality audio if the source does not follow a pitch contour similar to the training data. Moreover, conversion between instruments would be enhanced by the model ability to add specific modulations with respect to the target (e.g. adding vibrato when converting a flute to violin).
- ▷ Interactions for composition. The envelope rate at the input of these implicit acoustic models is relatively high (e.g. 250Hz) and unsuited for direct user control, an approach to allow composition would thus require to convert a score (e.g. pitch and velocity targets) into realistic and fine-grained acoustic envelopes for conditioning such neural audio synthesis models [134].

4.4 Learning Representations of Timbre

The previous section has detailed two main families of implicit timbre models. The first is based on domain translation which treats timbre as a categorical class and learns it by disentangling domain invariant features (content, e.g. pitch and velocity) from domain specific features (style, e.g. timbre). These translation models can process multiple timbres as implicit transformations of a given source sample to other target domains. The second is based on fine-grained fundamental frequency and loudness conditioning which lets the model learn a single timbre as the remaining acoustic features. For instance by hierarchical in-painting of the spectral distribution or by prediction of the corresponding harmonic and stochastic components. This approach enables independent manipulations of f0 and loudness, variable-length waveform synthesis, as well as timbre transfer to the learned instrument target. Yet it does not provide any direct control over continuous timbre variations which would allow to change the sound of the instrument at a fixed pitch and velocity.

Implicit timbre embeddings. An approach to perceptual drum sound synthesis is proposed in [215] which conditions a Wave-U-NET [251] architecture with semantic

descriptors from the Audio Commons extractor [4]. The chosen architecture is designed for waveform-to-waveform applications such as source separation and its adaptation to drum synthesis is done by providing an input loudness envelope up-sampled at the audio rate which is converted into a drum waveform corresponding to the global conditioning features of the semantic target. The feature extraction is based on a pretrained regression over human annotated audio for the attributes *hardness, depth, brightness, roughness, boominess, warmth* and *sharpness* which are provided as conditioning variables and allow shaping the target timbre via continuous semantic descriptors. However the relationship between semantic descriptors and instrument timbre (e.g. which target drum class) is not learned explicitly and these descriptors may seem in parts antagonist (e.g. brightness and warmth) or overlapping (e.g. hardness and boominess) which induces some entanglement in the controls. An approach to pitched note synthesis is proposed in the Symbol-to-Instrument Neural Generator (SING [52]) which pretrains a waveform auto-encoder and learns a latent feature regression model conditioned with pitch, velocity and instrument class targets. The bottom auto-encoder learns a frame-wise acoustic embedding and the upper latent regression model is a RNN which infers ordered series of frame features given the target note classes. This results in a controllable note sampler which combines the RNN and the decoder into a waveform synthesis pipeline from global attributes. The GAN framework is extended to conditional audio synthesis in [69] by providing the generator with categorical pitch and velocity attributes as one-hot vectors concatenated to the global latent prior. As opposed to implicit acoustic models which abstract timbre from envelopes of the local fundamental frequency and loudness, this global conditioning allows the GAN latent prior to model continuous timbre variations pertaining to a given combination of pitch and velocity targets. The generator outputs a spectrogram representation with both the magnitude and the instantaneous frequency which can be approximately mapped to the complex spectrogram and inverted with iSTFT. As a result, the model can synthesise audio with little latency and allows a continuous timbre control over the generation of individual notes of fixed duration. However, the GAN latent space is little interpretable since it does not correspond to an inference mechanism which would allow to analyse and visualise the embedded timbre distributions.

[4] `https://www.audiocommons.org/2018/07/15/audio-commons-audio-extractor.html`

Learned analysis/synthesis timbre representations. Early works on neural musical audio synthesis have used auto-encoders on magnitude spectrogram frames to learn a low-dimensional invertible audio representation which embeds timbre as well as other features (e.g. pitch) in an entangled manner [229]. As a result these features are being manipulated all together, which prevents from predictable transformations. A comparison of diverse auto-encoder models for reconstruction of short-term magnitude spectrograms of musical notes is conducted in [226] which highlights the benefits of variational auto-encoders in terms of usability of the continuous latent representation. Direct waveform modelling is introduced with a WaveNet auto-encoder in [71] along with the reference Nsynth dataset [5] of annotated musical notes. The encoder output is down-sampled in time by average pooling, yielding a variable length embedding that is concatenated with the categorical pitch target, up-sampled to the audio rate and used as conditioning to the WaveNet decoder that synthesises audio in an auto-regressive fashion. For that reason, the model is computationally intensive and unsuited for real-time inference, although an open source interface has been developed for controlling a GPU-equipped server back-end [6]. This model lets users morph instrument timbres by continuous interpolations in the embedding given the desired pitch target. A disentangled latent space of pitch and timbre is learned in [171] with a VAE on fixed-length magnitude spectrograms of sustained note portions. The encoder outputs parameters for two separate Gaussian distributions over pitch and timbre, which form a Gaussian mixture that can be independently sampled and jointly decoded. An extension of this model for unsupervised learning is proposed in [172], which proposes novel losses based on the assumption that small pitch shifts do not modify timbre. In this setting, ground-truth labels are not available to train latent classifiers and enforce that each latent space is discriminative with respect to either pitch or timbre. Thus unsupervised disentanglement is encouraged by first having a categorical output distribution for the pitch encoder and a continuous one for the timbre encoder. During training, an input audio is pitch shifted and a regression loss is applied to the timbre encoding which should be invariant to moderate pitch shifts. The additional losses for unsupervised disentanglement are based on cycle-consistency (decoding with a different timbre or pitch latent and re-encoding to the same latent), contrastive learning and a surrogate loss that regresses the known pitch shifting deviation in the pitch encoding. To evaluate the unsupervised disentanglement achieved by the model, a metric for pitch clustering

[5] `https://magenta.tensorflow.org/datasets/nsynth`

[6] `https://github.com/googlecreativelab/open-nsynth-super`

and a consistency-diversity score for conditional generation are introduced. Another approach to disentanglement of pitch and timbre is proposed in [254] using a conditional VAE on parameters of a source-filter representation. The harmonic component of the SMS [237] model is decomposed into fundamental frequency and cepstral coefficients (as features of timbre) which are both analysed with the encoder. The decoder uses the estimated fundamental frequency as conditioning information and inverts the learned latent space, which is expected to embed timbre, into the harmonic spectral envelopes. In contrast with the parametric model of DDSP [70], the stochastic residual component is not processed and the reconstruction is evaluated on the parameters of the source-filter model rather than on the output waveform, which inherently bounds the model quality to that of the pre-defined analysis/synthesis.

4.5 Score and Audio Processing

The previous sections discussed generative models focused on acoustic production, for instance synthesising individual audio events for some given pitch and velocity targets which is the function of a note sampler. Or synthesising an audio performance from fine-grained acoustic envelopes which implicitly carry the underlying music structure but do not allow direct composition from the symbolic domain. Expressive audio modelling from score raises the challenges of both the global structure understanding, for instance modelling the relationships between the notes played to generate a realistic melody articulation and chords, the intermediate control on the performance timbre and the local fidelity of the synthesis output. It also relies on the availability of aligned score and audio datasets, a ressource which is much harder to collect than individual sample audio libraries, or calls for unsupervised learning methods to tackle the complex task of bijective mapping between the audio and score domains [39]. For this purpose, the Wave2Midi2Wave model [106] is introduced along with the unprecedentedly large MAESTRO dataset[7] that was collected using the Yamaha Disklavier piano[8], an electromechanical device which automatically transcribes the played performance into score. The model is composed of three modules that are separately trained in order to factorize the different data processes involved in piano music modelling. The Onsets and Frames model [105] is trained for supervised piano transcription from audio

[7] https://magenta.tensorflow.org/datasets/maestro

[8] https://www.piano-e-competition.com/ecomp_yamaha.asp

to MIDI. Using this pretrained transcriber, it is possible to automatically annotate new piano performances and train a language model for symbolic music [118] which can generate new compositions that are musically consistent with the considered performance dataset. Given the discrete score conditioning (onsets from a MIDI piano-roll), a WaveNet synthesizer is trained to reconstruct the audio with high-fidelity. By combining state-of-art modules, the Wave2Midi2Wave system is able to generate piano music such that both the composition and the performance are perceptually similar to human-made music. However, its transcription is tailored to piano music and the rendering does not offer expressive acoustic controls over the performance. Moreover, due to the complexity of the separate models, it does not train end-to-end and is unlikely to adapt to smaller annotated datasets such as those available for other instruments. A similar large-scale dataset collection was carried for drumming performances [9] using the Roland TD-11 electronic drum kit [10], which enabled the supervised training of a reference model for drum transcription [30], yet there is no equivalent counterparts for other instruments such as strings or winds.

A score to audio pipeline for cello solo music generation is developed in [176] by training a bi-axial LSTM model in the symbolic domain to condition a down-sized WaveNet synthesizer pretrained with the annotated audio of the MusicNet database, about 50 minutes of recorded performances for that instrument. To alleviate the difficulty of training a waveform synthesis model with a limited amount of annotated data, other works have used a two step approach by training a score to spectrogram model which can be inverted with generic models such as a WaveNet vocoder pre-trained on unlabelled audio. The PerformanceNet [278] model generates high-quality spectrograms from score in a coarse to fine manner. The first module is a U-NET that converts the input piano-roll into a low-resolution acoustic representation that is up-sampled in the frequency dimension with a second module. The acoustic resolution is increased with a multi-band tree structure which doubles the number of frequency channels until the target spectrogram output size. The Mel2Mel [141] model focuses on timbral expressivity by conditioning the inputs and outputs of a bi-directional RNN with an instrument embedding. Instrument-dependent input hidden features are obtained from the score information, which are expected to represent the temporal envelopes and dynamics of instrumental notes. Mel-spectrograms are generated from the RNN outputs by a sec-

[9] `https://magenta.tensorflow.org/datasets/groove`

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

ond layer with instrument conditioning, in order to add the spectral information and acoustic features of the target timbre. The expressive conditioning from the instrument embedding is done by feature-wise linear transformation [204] such that a given melody can be played by different instruments and the learned timbres can be morphed. However, because annotated datasets with balanced instrument ratios were not available, the model is trained on synthetic audio produced from piano melodies by sample-based rendering in the different target instruments. Expressive control for piano performance synthesis is proposed in [257] with a gaussian mixture recurrent VAE that predicts Mel-Spectrogram with a latent representation that separately accounts for articulation and dynamics. Each expressivity parameter relating to either note duration or velocity is extracted from score onsets and binarised in between {staccato,legato} and {soft,loud} that are encoded as separate gaussian mixture components. This expressive encoding is provided to the decoder that generates spectrograms that are inverted to audio by a pretrained WaveGlow synthesizer. By sampling or morphing in the mixture components, it is possible to modify the style of the given score rendering. Another research direction for score to audio generation relies on pre-trained implicit acoustic models of a given target timbre [70] [186] which are controlled with f0 and loudness envelopes at an intermediate sampling rate. This highly-compressed acoustic representation allows to train light-weight control models [134] [33] between the quantised pitch and velocity targets and the corresponding natural envelopes of the target instrument articulation (Figure 22) while the neural synthesizer implicitly generates all the acoustic details of the target timbre.

4.6 Perceptual Audio Embeddings for Generative Modelling

Both the training and the evaluation of generative audio models can make use of pre-trained network embeddings, either in the form of a perceptual loss or as a basis for statistical scores such as the Fréchet Audio distance. Besides the specific network architectures, these embeddings differ by their training tasks and datasets which we may divide in two groups: generic classifiers trained at categorising many unrelated classes or specific embeddings trained at analysing a particular sound feature (e.g. pitch). Accordingly, the former group can provide a general metric in which many features may be entangled whereas the later can provide a distance which accounts for a single target feature.

Generic audio embeddings. Large scale audio event classification was introduced with the AudioSet [89] dataset which contains more than two millions audio clips from ten second long Youtube videos labelled into 632 classes of various kinds, including environmental noises, musical sounds and voice. A comparison study [109] was carried on this dataset to adapt the state-of-art CNN architectures from computer vision to supervised audio classification based on a common Mel-spectrogram input representation. One such example is the VGGish[11] adaptation of [243] to the acoustic classification of the AudioSet ontology. The VGGish model was used as the reference embedding for generative evaluation with the Fréchet Audio Distance [139] as well as for computing Gram losses for audio synthesis [6] by deep feature matching. A large-scale audio embedding is learned without labels in SoundNet [10] which uses the output features of a pretrained image classifier on video frames to match its own features extracted from the corresponding raw waveform audio. This training relies on the underlying audio-visual feature correspondence between video frames and aligned audio to train the audio embedding on unlabelled videos under the feature distribution of the pretrained image classifier. The experiment demonstrates the efficiency of the method for transfer learning by using this audio feature extraction to train a shallow classifier on limited amount of labelled audio, for instance 2000 clips of the ESC-50 dataset [206], which outperforms training a classifier from scratch only on the labelled target data. The Look, Listen and Learn (L3 [7] and OpenL3 [47]) method proposes a fully self-supervised approach to learn audio-visual embeddings by jointly training two parallel sub-networks either processing image frames or aligned audio spectrograms. The outputs of both networks are fed into a fusion layer which optimises a discriminative task by either classifying input pairs of the two modalities as corresponding or not. Because creating wrong pairs does not require any labels (e.g. randomly swapping audio or image frames between different videos), this method allows training both image and sound embeddings with large uncurated datasets. In the experiment of transfer learning for training a downstream supervised audio classifier, the openL3 embedding is shown to outperform both SoundNet and VGGish although its self-supervised pre-training is the least constrained in terms data requirements.

Feature-specific embeddings. The features extracted by the aforementioned large-scale embeddings are little interpretable since discriminating unrelated classes of Au-

[11] https://github.com/tensorflow/models/tree/master/research/audioset

dioSet or assessing audio-visual correspondences from uncurated videos requires the implicit learning of many underlying audio properties, some of which may relate to pitch, timbre or temporal dynamics in an entangled manner. For this reason they are efficient at providing a general pool of features for transfer learning to other downstream tasks, for instance training a classifier on a new and possibly unrelated dataset. On the other hand, some classifiers which are trained at predicting a specific feature can provide a more interpretable embedding which is expected to discriminate that particular property while being invariant to other ones. Considering a reference pitch detection model such as CREPE [142], we can expect its feature statistics to provide a distance closely related to pitch variations and more invariant to timbre since it learns to perform that task regardless of the many different training instruments. A CREPE embedding loss is used to train a self-supervised DDSP [70] model by providing a learning signal which specifically assesses the model performance at generating the correct fundamental frequency envelope. This technique is also used in [186] to ensure that the iterative wave-shaping preserves the input pitch contour while adding the desired timbre distribution. The learning of a differentiable metric of perceived audio fidelity is introduced in [175] which uses human labels to assess whether various strengths of audio distortion are perceptually noticeable or not, in order to train a classifier at the threshold of just noticeable differences (JND). Audio pairs are presented to the listening test with some randomised audio perturbations (e.g. noise, equalisation, compression, reverberation) and rated as whether they sound exactly the same or not. The model is comprised of a convolutional feature extractor, with activations of the l-th layer denoted as $F_l(.) \in \mathbb{R}^{T_l * C_l}$, which yields a deep feature distance:

$$D(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{l=1}^{L} \frac{1}{T_l * C_l} ||\mathbf{w}_l \odot (F_l(\mathbf{x}) - F_l(\hat{\mathbf{x}}))|| \tag{61}$$

computed with a learnable channel weighting $\mathbf{w}_l \in \mathbb{R}^{C_l}$. This distance is learned as part of a JND classifier E which is trained at predicting the label $y = \{0, 1\}$ of whether the clean audio $\mathbf{x}$ and perturbed audio $\hat{\mathbf{x}}$ are perceptually dissimilar $y = 1$ or indistinguishable $y = 0$. Accordingly, the convolutional feature extractor and shallow classifier jointly optimise the following binary cross-entropy

$$\mathcal{L}(F_{l=1...L}, E) = \text{BCE}(E(D(\mathbf{x}, \hat{\mathbf{x}})), y). \tag{62}$$

Since the model classifies ground-truth human ratings, its learned distance accounts for the perception of audio quality and outperforms classical metrics of audio quality by having a stronger correlation with blind mean opinion scores. Moreover, as the distance is computed in a neural network embedding it can be used as a differentiable loss to train down-stream tasks such as audio enhancement models (e.g. speech denoising).

5 Experiments in Neural Audio Synthesis

In this section, we present the results of the experiments carried and published during this thesis which have been framed into three main topics. These topics are implicit modelling of timbre, learning representations of timbre and light-weight neural audio processing. For each experiment, we provide a global introduction and discussion of the main contributions that is followed with details and quantitative results. These are provided in the corresponding papers, following the PhD thesis by publication model, and additional contents are linked in the project webpages. To ease reading, published papers are reformatted in a common template style and links to the original templates are provided.

5.1 Implicit Timbre Models

As witnessed in the image domain, generative modelling techniques have proven successful at converting data between semantic domains with applications to processing visual artistic styles [304]. Unpaired domain translation models enable generation driven by example, provided a source sample and the learned mapping to another target domain, these models transform parts of the input features to match the specific distribution of those observed in the output domain. For audio applications, this motivates our experiment of using domain translation as a model of timbre transfer which considers sample libraries of different instruments as domains across which we would like to convert features that belong to the relative perception of each acoustic source. The concrete application of this process resembles that of an audio effect which performs cross-synthesis by mixing the user-provided input sound sample with auditory properties learned in the target domain, for instance we would like to transform a violin note as played by a trombone.

For this task, we choose the Studio-on-Line dataset [12] which is a library of individual note recordings across the whole tessitura of the *Piano, Cello, Violin, Flute, Clarinet, Trombone, French-Horn, English-Horn, Oboe, Saxophone, Trumpet* and *Tuba*. Each instrument is recorded in several dynamics and playing styles, some of which are usually specific to an instrument or a family (e.g. pizzicato for the strings), so that multiple variations are available for a same pitch. We down-sample the audio

to 22050Hz, normalise amplitudes and compute log-magnitude spectrograms with the Non-Stationary Gabor Transform (NSGT [11]) with 500 bins on the Mel frequency scale. Each spectrogram is sliced into blocks of 16 frames so that the model inputs are of shape (500,16). We choose this representation for the log-scaled frequency axis which increases the separation of lower harmonic frequencies, its efficient approximate inversion with the Griffin-Lim algorithm and the compactness of the input which we wish to visualise in a 3-dimensional latent space. We first validate this setting by training a VAE model on the database of multiple instruments, which can be done without conditioning or by using additional database labels corresponding to either pitch or instrument classes. Accordingly, we can observe how the distribution of instruments in the latent space varies and that the instrument conditional VAE has the least separated class projections.

In the second place, we adapt this spectral VAE baseline to the UNIT [164] framework which performs bijective domain translation using a pair of VAEs and corresponding domain-specific adversarial networks. The translation model relies on the shared latent space assumption which states that two hypothetically matched samples in both domains should map to the same latent code that would then be equally decoded back to both domains. This is enforced by weight sharing in the upper network layers, domain specific adversarial learning and cycle-consistency regularisation so that the input of one VAE can be retrieved from the output of the other VAE. Because the model learns a continuous latent space shared for both domains, it allows translation as well as generating some variations by moving in the 3-dimensional space while decoding in the two output domains. In order to enhance the model control, we train a UNIT variant with pitch conditioning so that we can either choose some transposition or set the translation output pitch to that of the input. One major limitation of the UNIT framework is that it does not scale efficiently to more than two domains, as it would require to train an increasing number of losses (6 for the bijective case) and adversarial discriminators. One workaround that we investigated is to chain multiple UNIT models, for instance given the learned mappings $\mathcal{X}_1 \leftrightarrow \mathcal{X}_2$ and $\mathcal{X}_2 \leftrightarrow \mathcal{X}_3$ we could approximately convert $\mathcal{X}_1 \leftrightarrow \mathcal{X}_3$ by feeding the output of the first translation as input to the second, although we compound errors of both models that were trained separately.

This work was submitted to the first international conference on timbre organised by McGill University in Montreal, Canada and presented as a poster.

Timbre transfer between orchestral instruments with semi-supervised learning

Adrien Bitton, Axel Chemla-Romeu-Santos & Philippe Esling

(original publishing template for abstract and poster available at
`https://www.mcgill.ca/timbre2018/files/timbre2018/timbre2018_proceedings.pdf`
`https://github.com/acids-ircam/Timbre_MoVE/blob/master/docs/poster_2018.pdf`)

Motivations

We aim to provide new ways of synthesizing timbres by high-level interaction and transfer of properties between instruments. Our hypothesis is that each instrument defines a timbral domain.

Challenges in prior timbre studies. Analysis spaces of perceptual ratings are not invertible. They do not generalize to new audio nor they allow synthesis. Automatically extracted audio descriptors show limited correlations to timbre spaces. Representations based on DSP have a high number of parameters and analysis dimensions which require additional knowledge models to interprete and manipulate timbre.

Our proposal. We apply variational learning for finding high-level structured representations by joint optimization of analysis and generation processes. Dimensionality reduction yields 3-dimensional latent spaces of higher-level abstraction which are shared across multiple instrument domains and organized without need for human ratings.

Machine Learning Background

Variational Auto-Encoder. Modeling the data distribution $p(\mathbf{x})$ based on a lower-dimensional latent representation $\mathbf{z}$ that retrieves $\mathbf{x}$ so that $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. Approximate solution through variational inference (VAE [145]) over parametric families of candidate distributions for the encoder q_ϕ and the decoder p_θ that are optimized on the evidence lower bound (ELBO). This generative model is fast to train and effective on small datasets.

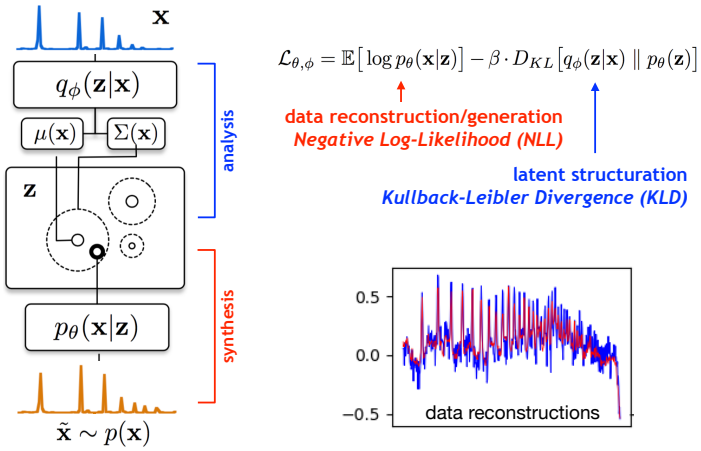

Figure 23: Auto-encoder model and training objective (ELBO) for reconstruction of spectrogram frames. The spectrum (blue, normalised amplitude) is analysed and reconstructed (red).

Unsupervised Translation Networks. Paired data domains are mapped onto a common knowledge space in the UNIT [164] framework. The underlying hypothesis is a shared latent space that would allow transfer from one domain to the other. It does not rely on matched samples across domains, instead, an additional adversarial objective pushes separate decoder layers to match their domain attributes from any latent coordinates. This leads to a competitive optimization by which the generator produces more realistic samples while the discriminator becomes more accurate at detecting fake samples. Domain translation is performed by switching decoders. A cycle-consistency regularization is applied so that the reversed transfer should reconstruct the original sample.

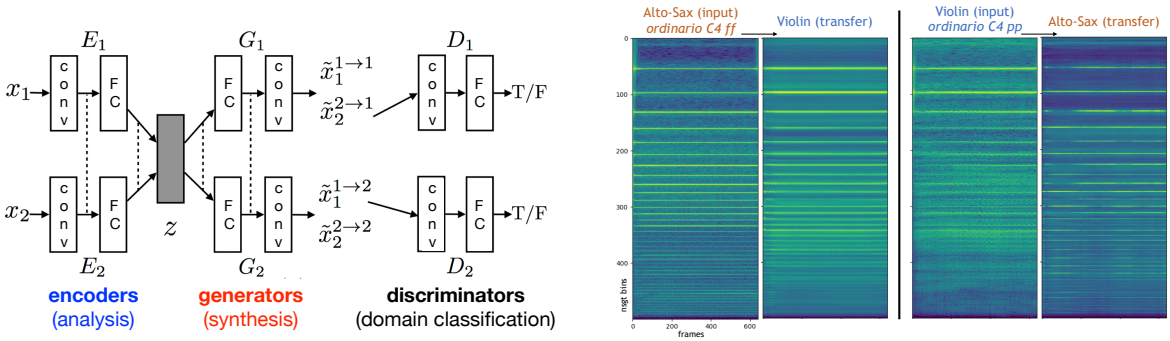

Figure 24: The UNIT model implementation for timbre transfer on NSGT magnitude spectrograms. The networks are composed of convolutional layers (conv) and fully-connected layers (FC). The paired auto-encoders map to a common latent space and use shared FC layers. On the right is shown an example of bi-directional transfer between Violin and Alto-Saxophone.

Timbre transfer with generative models

Implementation details. We use the Studio On Line dataset (SOL [12]) which comprises note recordings of 12 orchestral instruments (winds, strings, keyboard, brass) in several dynamics, playing styles and all pitches. We pre-process audio by Non-Stationary Gabor Transform (NSGT [11]) with Mel scaled frequencies as model input for its beneficial representation and invertibility properties. The encoder and decoder are built with 2D-convolutional and transposed layers to process blocks of NSGT-Mel frames with a 120 milliseconds context.

Timbre transfer strategies. Instrument-conditional VAEs and translation by switching the encoding condition to any decoding target condition. The model learns latent sub-spaces for each instrument condition. UNIT-like translators between domains (single instrument each) and translation by switching decoder from one domain to the other. The model learns a shared latent space for one-to-one transfer with possible semitone conditioning to control transposition over transfers.

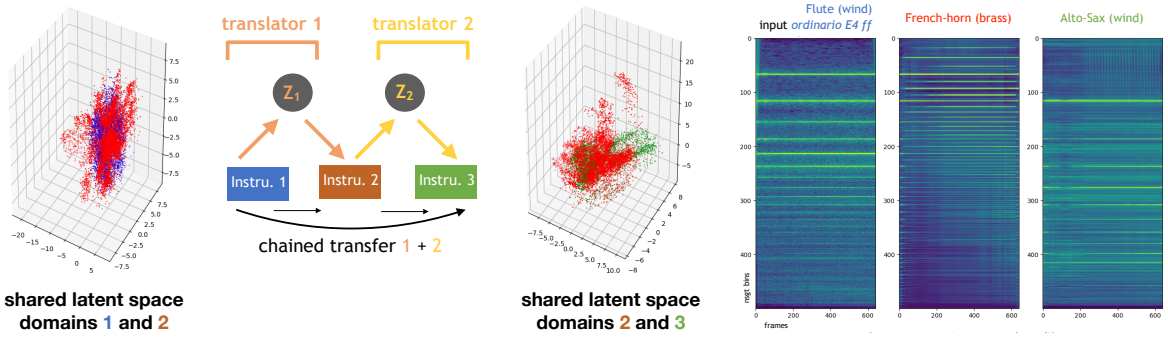

Figure 25: Pairing of two UNIT translators to translate across multiple domains using a common domain. On the left are visualized the two shared latent spaces, on the right are shown consecutive transfers from Flute to French-Horn (shared domain) to Alto-Saxophone.

Timbre processing by domain translations. The UNIT framework relies on unsupervised learning of domain-specific features with adversarial networks to match each decoder outputs with the observed data distributions of their respective target domains. This conveniently removes the need for paired data by which the conversion objective could be computed as a reconstruction error. However, this implies training one dedicated discriminator per domain and incurs the potential optimisation instabilities often observed in the adversarial matching of high-dimensional data distributions. An alternative unsupervised objective can be formulated with non-parametric two-sample test statistics such as the Maximum Mean Discrepancy (MMD [97]), which has been applied to generative moment matching [67] [162]. In our setting, we consider the spectrogram outputs of the decoder and the random samples from the target domain as two distributions which are approximated with the kernel density estimator (e.g. radial basis function) from which the MMD distance is computed. As opposed to the GAN framework, the MMD provides a fixed estimator which can be computed independently across domains and sub-networks, making it efficient for multi-domain translation. Although we did not explore this option to keep our method scalable to many domains, an in-between approach was proposed by training parametric MMD kernels as learnable discriminators [161].

We keep the same data settings as in the previous timbre translation experiment and investigate the extension of the UNIT framework for many-to-many domain translation. In the first place, we run a modified UNIT timbre transfer experiment by replacing the two adversarial losses with MMD distances computed in each data domain. Then we replace the VAE pair by an instrument conditional VAE which is trained at processing all source and target domains in a single architecture, which alleviates the second limitation of scalability observed in the UNIT framework. The processing of all domains in a single VAE model requires it to learn to modulate analysis and synthesis features with respect to the different instrument conditions, as opposed to train a separate sub-network for each domain. We implement the conditioning by Feature-wise Linear Modulation (FiLM [204]) on the normalised hidden activations using biasing and scaling parameters predicted by a conditioning sub-network. The overall architecture is jointly trained with the VAE objective in the randomised source domain conditions (reconstruction), the MMD matching with randomised target domain conditions (conversion) and the cycle-consistency loss (regularisation).

The bijective timbre transfer task is evaluated on several model variations, from the default UNIT framework that we iteratively modify by adding the concatenative pitch conditioning, the MMD loss (instead of GANs), the conditioning with FiLM layers and the single VAE model with instrument domain conditioning. We observe that the UNIT models with separate VAEs perform best in terms of reconstruction but that the proposed single VAE model performs best in terms of statistical metrics of transfer (MMD and k-NN clustering). We also visualise the topology of acoustic descriptors in the target domain ground-truth samples and synthetic conversion samples which confirm that the acoustic distributions are matched. Finally, we show results of training the model for many-to-many domain translations on four instrument classes: Saxophone, Flute, Violin and French-Horn. Some additional visualisations and audio samples are hosted on the dedicated online repository: `https://github.com/acids-ircam/Timbre_MoVE/`.

Further experiments on many-to-many timbre conversion could be carried using a conditional GAN framework such as Star-GAN [42] which alleviates the need of training multiple sub-networks. Another approach to be investigated could rely on non-adversarial statistics matching (e.g. moment matching) with disentangled deep feature embeddings. Neural network pretraining on specific tasks and data can provide differentiable perceptual losses for training generative models, for instance the CREPE pitch embedding or the JND embedding of perceived audio quality [175]. If multiple embeddings do not overlap in terms of discriminative features, they could be used to compute independent distances in the source and target domains without paired data, for instance preserving the input pitch while matching an output timbre distribution. This approach has the advantage of being potentially more stable and less demanding in computations as only the generator is trained. Moreover it provides interpretable losses with respect to the pretrained features, as opposed to the learned discriminator which may fail at capturing certain modes of the distributions or learn a trivial policy.

Modulated Variational auto-Encoders for many-to-many musical timbre transfer

Adrien Bitton, Philippe Esling & Axel Chemla-Romeu-Santos
(original publishing template and appendix available at
`https://arxiv.org/abs/1810.00222`)

Generative models have been successfully applied to image style transfer and domain translation. However, there is still a wide gap in the quality of results when learning such tasks on musical audio. Furthermore, most translation models only enable one-to-one or one-to-many transfer by relying on separate encoders or decoders and complex, computationally-heavy models. In this paper, we introduce the Modulated Variational auto-Encoders (MoVE) to perform musical timbre transfer. We define timbre transfer as applying parts of the auditory properties of a musical instrument onto another. First, we show that we can achieve this task by conditioning existing domain translation techniques with Feature-wise Linear Modulation (FiLM). Then, we alleviate the need for additional adversarial networks by replacing the usual translation criterion by a Maximum Mean Discrepancy (MMD) objective. This allows a faster and more stable training along with a controllable latent space encoder. By further conditioning our system on several different instruments, we can generalize to many-to-many transfer within a single variational architecture able to perform multi-domain transfers. Our models map inputs to 3-dimensional representations, successfully translating timbre from one instrument to another and supporting sound synthesis from a reduced set of control parameters. We evaluate our method in reconstruction and generation tasks while analyzing the auditory descriptor distributions across transferred domains. We show that this architecture allows for generative controls in multi-domain transfer, yet remaining light, fast to train and effective on small datasets.

Introduction

Music information can be analyzed in many forms, each of which conveys different specificities over musical qualities. Among these, *timbre* is the set of properties that distinguishes one instrument from another playing at the same pitch and loudness. Timbre has become a core concept in music composition since the 19^{th} century [180]. It has been studied using human dissimilarity ratings to construct *timbre spaces*, which exhibit the perceptual relationships between instruments [98]. However, these spaces are not invertible to the signal domain and do not generalize to new examples [181]. The heavy reliance on hand-crafted audio descriptors

to analyze timbre perception altogether leads to a lack of established models to understand and generate timbres [180]. Moreover, the specific nature of music tasks requires tailored evaluation principles that are yet to be ascertained [130].

Recent advances in *generative models* open alternative avenues to analyze highly dimensional data and tackle complex subsequent tasks. Amongst these, the idea of *style transfer* [88] has recently gained a flourishing interest. This approach allows to modify the stylistic features of an image while preserving its overall content and led to the more generic question of *domain translation*. In the recent UNsupervised Image-to-image Translation (UNIT) model, [164] proposed to learn a shared latent space with a Variational Auto-Encoder (VAE) and translate between different data domains with an adversarial criterion. However, specific properties of the generation cannot be controlled and that discriminative objective might lead to an unstable and longer training. Here, we first extend this approach to musical transfer while improving it by introducing Modulated Variational auto-Encoders (MoVE) that offer control over the generation through conditioning. Furthermore, by replacing the discriminative networks by a Maximum Mean Discrepancy (MMD) objective, we alleviate the need for an additional adversarial training specific to each domain.

Although UNIT provides a powerful framework, it only applies to *one-to-one* transfer. This implies that a different model has to be trained for each pair of domains. To mitigate this issue, [42] proposed StarGAN which performs *many-to-many* transfer between several domains. However, it relies solely on Generative Adversarial Networks (GANs) and does not learn an implicit task representation to interact with. In the music realm, [190] proposed Universal Music Translation (UMT), which does not use GANs. Although it enables translation across multiple complex audio domains, this method requires to learn a separate decoder for each domain, which leads to a prohibitive training time. In contrast to these methods, we show that MoVE can be further conditioned on domain information and generalizes to *many-to-many* transfer with a single encoder and decoder architecture able to perform multi-domain transfer. The resulting models are rather lightweight and fast to train while effective on a moderate amount of examples.

Here, we define *timbre transfer* as applying a variable part of the auditory properties of a musical instrument onto another. We circumvent the lack of definition for timbre by considering each instrument as a separate domain that maps onto a common latent representation. We further address the crucial need for interactivity and control in creative applications such as audio synthesis. Accordingly, our method yields 3-dimensional latent spaces that can be explored and controlled through high-level explicit variables such as pitch and octave values. This supports sound generation with smoothly evolving timbre qualities and complex

domain transfers from a reduced set of parameters. Finally, we analyze traditional audio descriptor distributions when transferring between multiple domains or decoding across latent dimensions to demonstrate the generative capacities of our model.

Related works

Style transfer and image translation. In computer vision, style transfer [88] has been proposed to generate images that preserve the content from a source image but feature stylistic qualities belonging to another target image. Although this technique provides compelling results, it operates on local textural information and fails to capture higher-level semantic properties of the style. Further research has been carried to address this question of *domain translation*, first proposed by [129]. In the fully supervised setting, this translation would require paired samples. However, such datasets are scarce and the concept of existing samples exactly matching the translation task is restrictive from a generative perspective. In the UNIT approach [164], the underlying assumption is that two hypothetically matching samples should map onto the same point in a shared latent space. Hence, translation is achieved by partially weight-shared VAEs in order to map the two separate domains to a common latent representation. Learning is performed with an auxiliary pair of adversarial discriminators which push translated samples to match the distributions of their respective domains. An additional *cycle-consistency* objective [304] reinforces the shared learning by ensuring that translated samples can be retrieved back to their original domains. However, this architecture can only operate on single domain pairs.

In order to provide *many-to-many* translations, [42] proposed to replace weight-sharing by conditioning a single GAN. This allows to train on multiple domains simultaneously, while enabling control over the generative process. However, the authors evaluate only on highly similar domains (eg. face attributes). Furthermore, this approach relies solely on GANs, which are notoriously difficult to train, prone to lack full support over the data [102] and do not provide a latent space encoder.

Recently, Feature-wise Linear Modulation (FiLM) has been proposed to improve conditioning by learning conditional bias and scaling throughout a network [204]. This method was successfully used in image stylization [90] where adaptive modulation conditioned on a style image is applied after each intermediate instance normalization. Here, we show that by relying on FiLM layers for domain conditioning, we can perform *many-to-many* domain translation with a single VAE architecture, as depicted in Figure 26. Moreover, by using a MMD criterion, we alleviate the need for GANs or specific adversarial discriminators. Hence, we

obtain an unsupervised, lightweight and easy to train model with a general and controllable latent space.

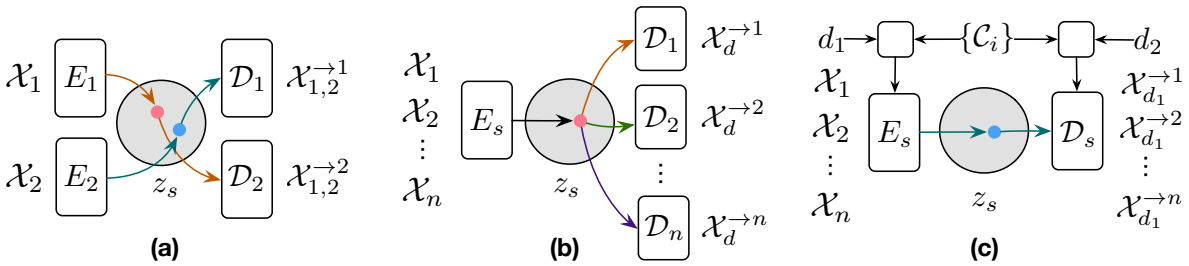

Figure 26: Different approaches to domain translation. (a) One-to-one transfer models such as UNIT are restricted to domain pairs. (b) One-to-many transfer models such as UMT require to train a different decoder for each domain. (c) Our proposed many-to-many transfer model (MoVE) allows to perform multi-domain transfer with a single encoder and decoder, while providing control over the generation with external high-level conditioning variables.

Audio translations. Recent applications of generative models to audio have shown promising results, notably supported by solutions to efficiently generate waveforms such as Wavenet [268] or SampleRNN [185]. Most of these proposals target voice signals and there is still a large gap when addressing musical data. Some approaches have tackled musical style transfer [274]. However, as pointed by [51], musical style is a multimodal and multi-scale notion, which implies a variety of underlying factors. Specifically for domain translation, [190] proposed a Universal Music Translation (UMT) network that globally translates musical recordings between different genres and instrument domains. Using a single Wavenet encoder and separate decoders for each domain, this approach is able to transform a given melody so that it is played by different instruments. By design, this method requires to train a specific Wavenet decoder for each of the target domain. It does not provide control over audio synthesis and the learned representation does not allow direct visualization nor transfer of only specific parts of timbre attributes. Hence, it does not enable informed generative processes, musical interaction and creativity. Our proposal targets 3-dimensional latent spaces supporting timbre transfer and continuous synthesis paths with explicit control over musical attributes.

Musical Timbre Transfer

Musical timbre can be defined as the *set of auditory qualities* that distinguishes two instruments playing the same note at the same loudness. Seminal studies relying on human dissimilarity ratings provided an interesting step towards understanding music perception [182].

However, the ordination techniques used yield non-invertible and fixed timbre spaces. Hence, they do not support audio synthesis nor do they provide a way to manipulate timbre structures. Signal processing techniques have also been developed to process and alter timbre. However, these rely on complex analysis schemes that decompose sounds into large sets of parameters [237], precluding intuitive control over the audio synthesis process.

Here, we propose to use generative models in order to perform musical timbre transfer. In order to circumvent the complexity of defining timbre, our underlying hypothesis is that each instrument defines a timbral *domain,* which contains all style qualities that shape its identity. Musical timbre transfer can be achieved by transforming a certain amount of the auditory features of a musical instrument according to another (eg. like playing a saxophone with a bow). Transferring all timbre properties of an instrument leads to *domain translation*, while partial modification of these amounts to *style transfer*. Furthermore, our goal is to obtain a controllable model that can be used for creative purposes. Hence, we aim to obtain 3-dimensional latent spaces along with high-level musical parameters that enable human interaction and control over the generation.

This type of transfer can be performed in several ways, as depicted in Figure 26. First, *one-to-one* transfer models such as UNIT map samples from a given pair of domains to a shared latent space. By learning separate layers and weight-shared layers in both the decoder and encoder, domain translation can be assessed through adversarial discriminators. We first adapt this model to timbre transfer and show that we can alleviate the need for GAN training by using an alternative MMD objective. It leads to a faster and more stable learning that we further enhance by modulating shared layers with FiLM layers [204] on pitch and octave. This provides an explicit control over generation, altering or not the pitch regardless of timbre. The *one-to-many* transfer models (UMT) allow to work with multiple domains but require to learn a different decoder for each. This leads to a more complex and longer training and reduces the generalization ability of the model gained through multi-task learning. Here, we show that our proposed Modulated Variational auto-Encoder (MoVE) allows to perform *many-to-many* transfers with a single VAE simultaneously processing all domains. The success of our solution relies on an efficient domain conditioning, together with external control variables, performed through FiLM layers acting on the whole network. This solution offers a greater generalization power by jointly learning all transfer tasks within a single architecture. The resulting latent space successfully models joint and conditional distributions over several instrument domains. This also enables control with semantic labels, while providing interactive 3-dimensional spaces to synthesize novel tones from a reduced set of control parameters.

One-to-one transfer

Our *one-to-one* transfer model is based on an architecture similar to UNIT [164] where the core idea is to learn a latent space that is shared between two domains $\mathcal{X}_1$ and $\mathcal{X}_2$. Based on samples $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$, we aim to model the joint distribution $p_{\mathcal{X}_1,\mathcal{X}_2}(\mathrm{x}_1, \mathrm{x}_2)$ over the two domains. By learning domain-specific encoders E_1 and E_2, matching samples drawn from each marginal distribution $p_{\mathcal{X}_1}(\mathrm{x}_1)$ and $p_{\mathcal{X}_2}(\mathrm{x}_2)$ should map onto the same latent code $\mathrm{z} = E_1(\mathrm{x}_1) = E_2(\mathrm{x}_2)$. Equally, any latent code can be decoded back to any of the two domains $d \in \{1, 2\}$ by learning appropriate decoders $\mathrm{x}_d^* = D_d(\mathrm{z})$. A paired VAE implements this assumption through separate domain-specific layers $\{e_d \ ; \ d_d\}$ alternated with weight-shared ones $\{e_{\mathrm{ws}} \ ; \ d_{\mathrm{ws}}\}$. The full encoders and decoders are defined by the composition of both parts $E_d = e_{\mathrm{ws}} \circ e_d$ and $D_d = d_d \circ d_{\mathrm{ws}}$.

Each VAE is trained with a reconstruction loss on its own domain, by approximating the intractable latent conditional $p(\mathrm{z}|\mathrm{x})$ with a parametric encoding network $q_\phi(\mathrm{z}|\mathrm{x})$ with $\phi \in \Phi$. In comparison to UNIT, we both use a Gaussian encoder q_ϕ and decoder p_θ so that $\mathrm{z} \sim q_\phi(\mathrm{z}|\mathrm{x}) = \mathcal{N}(\mu_\phi(\mathrm{x}), \sigma_\phi(\mathrm{x}))$ and $\mathrm{x} \sim p_\theta(\mathrm{x}|\mathrm{z}) = \mathcal{N}(\mu_\theta(\mathrm{z}), \sigma_\theta(\mathrm{z}))$. Training the model amounts to optimize $\{\theta \ ; \ \phi\}$ on the Evidence Lower Bound Objective (ELBO), defined as a Negative Log-Likelihood (NLL) term on the output prediction error and a β-weighted Kullback-Leibler Divergence (KLD) term that assesses the error from the approximate latent density against the intractable true posterior distribution.

$$\mathcal{L}_{\theta,\phi}^{rec.} = \mathbb{E}_{q_\phi(\mathrm{z})}[\log p_\theta(\mathrm{x}|\mathrm{z})] - \beta * D_{\mathrm{KL}}[q_\phi(\mathrm{z}|\mathrm{x}) \| p_\theta(\mathrm{z})]$$

This inference objective allows to learn structured low-dimensional and invertible representations of the data, while disentangling generative factors in the encoded variables [111].

Translation is performed by switching domains between the encoding and decoding stages eg. $\mathrm{x}_{1\to2} = D_2 \circ E_1(\mathrm{x}_1)$. However, there is usually no matching sample x_2^* that could allow to perform the optimization of the reconstruction error $\mathcal{L}_{\theta_2,\phi_1}^{1\to2} = err(\mathrm{x}_{1\to2} \| \mathrm{x}_2^*)$. To circumvent this challenge, UNIT relies on GANs to discriminate the generated translations against the target data distributions they model. However, this GAN criterion leads to a more complex and possibly unstable training process. Here, we show that we can efficiently replace the adversarial criterion by a differentiable distance measure on the probability distribution spaces. We minimize the Maximum Mean Discrepancy (MMD), a non-parametric kernel method [97], between the set of transferred samples $\mathrm{x}_{1\to2} \sim p_{\theta_2}(\mathrm{x}|\mathrm{z}, \phi_1)$ and a randomly sampled set from

the target domain $\bar{\mathrm{x}}_2 \sim p_{\mathcal{X}_2}$

$$\mathcal{L}^{1\to2}_{\theta_2,\phi_1} = \mathrm{MMD}[\mathrm{x}_{1\to2}\|\bar{\mathrm{x}}_2] = \mathbb{E}_{\mathrm{x},\mathrm{x}'}[k(\mathrm{x},\mathrm{x}')] - 2 * \mathbb{E}_{\mathrm{x},\bar{\mathrm{x}}}[k(\mathrm{x},\bar{\mathrm{x}})] + \mathbb{E}_{\bar{\mathrm{x}},\bar{\mathrm{x}}'}[k(\bar{\mathrm{x}},\bar{\mathrm{x}}')]$$
$$\forall\{\mathrm{x},\mathrm{x}'\} \in \mathrm{x}_{1\to2} \text{ and } \forall\{\bar{\mathrm{x}},\bar{\mathrm{x}}'\} \in \bar{\mathrm{x}}_2$$

where k is a Radial Basis Functions (RBF) kernel $k(\mathrm{x},\mathrm{x}') = \sum_{i=1}^{n} \exp^{-\alpha_i\|\mathrm{x}-\mathrm{x}'\|^2}$.

Reconstruction and translation objectives are jointly optimized with an extra cycle-consistency (CC) criterion. It consists in encoding a translated sample back to the latent space and decoding it to its source domain so that $\mathrm{x}_{cc1} = D_1 \circ E_2(\mathrm{x}_{1\to2})$. Hence, this double translation should retrieve the initial sample and the reconstruction error can be optimized with a NLL loss.

$$\mathcal{L}^{cc1}_{\theta_1,\phi_2} = \mathbb{E}_{q_{\phi_2}(\mathrm{z}|\mathrm{x}_{1\to2})}[\log p_{\theta_1}(\mathrm{x}|\mathrm{z})]$$

Finally, the complete optimization objective is defined as

$$\mathcal{L}^{train}_{\theta,\phi} = \mathcal{L}^{rec1}_{\theta_1,\phi_1} + \mathcal{L}^{rec2}_{\theta_2,\phi_2} + \lambda_{\mathrm{MMD}}(\mathcal{L}^{1\to2}_{\theta_2,\phi_1} + \mathcal{L}^{2\to1}_{\theta_1,\phi_2}) + \lambda_{\mathrm{CC}}(\mathcal{L}^{cc1}_{\theta_1,\phi_2} + \mathcal{L}^{cc2}_{\theta_2,\phi_1})$$

where λ_{MMD} and λ_{CC} allow to weigh the relative influence of different objectives. For the purpose of controllable musical timbre transfer, we further apply conditioning at the input of the weight-shared networks by concatenating one-hot encoded pitch classes and octaves. This pushes the shared encoder to structure note-agnostic features, while providing control over the generation.

Many-to-many transfer

In order to alleviate the *one-to-one* limitation that requires a different training for each domain pair, we propose the single MoVE architecture as depicted in Figure 27. All layers are shared over the multiple domains processed, by learning a single modulated encoder E_s and decoder D_s. Transfer is performed by switching the categorical condition between different instruments. Hence, the practical success of this method highly depends on the conditioning strategy, which must also retain the pitch and octave control. To do so, we use an input embedding that jointly maps these categorical conditions to dense vectors fed into FiLM generators. We replace each intermediate batch normalization with instance normalization and activation is followed by a FiLM modulation layer (conditional instance normalization). Biasing and scaling are either applied feature-wise for 1-dimensional activations or channel-wise after 2-dimensional feature maps. A different generator output is used for modulating each

instance normalization layer depending on its shape. The MoVE model trains in reconstruction with the ELBO and in transfer with the MMD, which is separately computed for each instrument against each of the others.

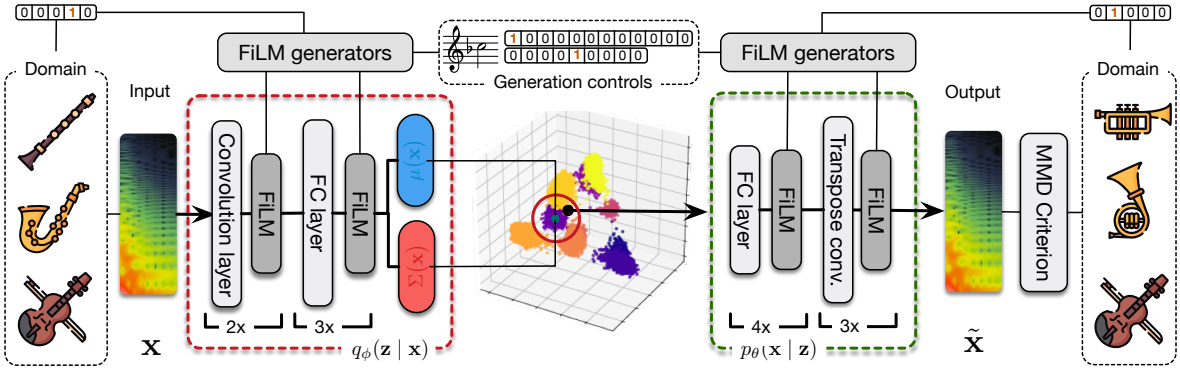

Figure 27: The Modulated Variational auto-Encoder (MoVE) provides a single architecture able to perform many-to-many transfer while controlling the generation with external parameters. Both the domain and control information are processed to modulate different layers of the architecture.

Experiments

Dataset. In order to learn our timbre transfer models, we rely on the Studio-On-Line (SOL) database of orchestral instrument note recordings [12]. We selected 12 instruments across the 4 families of *wind* (Alto-Saxophone, Bassoon, Clarinet, Flute, Oboe), *brass* (English-Horn, French-Horn, Tenor-Trombone, Trumpet), *string* (Cello, Violin) and *keyboard* (Piano). We consider each instrumental subset as a timbral domain $\mathcal{X}_i$, which contains the full tessitura of each instrument at different velocities (amounting to around 100 to 200 samples per domain). We split these subsets into 90% training notes and 10% test set. The audio waveforms are down-sampled to 22050Hz before computing the Non-Stationary Gabor Transform (NSGT) [11]. This spectral transform allows to map to a perceptual pitch scale, while remaining iteratively invertible to the signal domain [205]. NSGTs are computed on a scale of 500 Mel bins ranging from 10Hz to 11000Hz. The resulting matrix data is sliced into chunks of 16 temporal frames, amounting for a context of about 120ms. This yields a final input size of 16x500 dimensions. We keep only the magnitude information and lowest values are floored to $6e^{-5}$ before applying a logarithmic transform. Finally, we normalize the entire dataset by computing a zero-mean unit-range normalization on all training samples.

Implementation details. The *one-to-one* transfer network is implemented as follows. The first encoding stacks e_d are domain-specific, each composed of two 2-dimensional strided convolutions, an intermediate flattening step and a fully-connected (FC) layer. The intermediate representation is either concatenated (denoted as C-po models) with the conditioning vector of size 21 (12 pitch and 9 octave classes) or modulated by FiLM (denoted as F-po models). Follows a weight-shared set e_{ws} of two FC layers and two Gaussian encoder output layers mapping the input to a shared 3-dimensional latent space. All intermediate layers are followed by batch normalization and a Leaky-ReLU non-linearity. This structure is mirrored in the decoders, where the latent code is conditioned and fed into a weight-shared block d_{ws} of 3 FC layers. Follow separate decoding stacks d_d each with a FC layer, two transpose convolutions that up-sample the representation and two Gaussian decoder outputs. The final output activation is a Tanh applied to the decoded means, according to the initial data scaling. Full details of the architectures are given in appendix A.

The *many-to-many* transfer model relies on the same architecture, but without domain-specific encoders and decoders. Hence, all layers are similar but a single network jointly processes all domains thanks to FiLM layers (denoted as F-pod models). For these, an embedding layer maps our categorical vocabulary of pitch, octave and instrument classes to dense vectors which are processed by two FiLM generators, one for the encoder and one for the decoder. Each has 3 FC layers followed by scaling and biasing output pairs that each maps to the size of the modulated layer. We replace every batch normalization by instance normalization and apply FiLM generator outputs as linear transforms modulating normalized hidden activations. Conditional instance normalization is performed feature-wise for 1-dimensional vectors and channel-wise for 2-dimensional feature maps.

Regarding optimization, all training objectives are simultaneously back-propagated in all networks. We use a Xavier weight initialization and the ADAM optimizer with an initial learning rate of $1e^{-4}$. Following the β-warmup procedure [246], only the NLL reconstruction objective is optimized in the first epochs and the KLD strength is gradually increased from 0 to 1 during half the total number of training epochs. Similarly, we introduce the translation objective after 40 epochs and the optional cycle-consistency objective after 60 epochs. We train on mini-batches of size 128 and the MMD is evaluated against batches of size 2048 sampled from the target distributions and computed with three Gaussian kernel parameter values $\{0.05, 0.1, 1\}$. We found the magnitude of MMD gradients to be much smaller than that of the ELBO. Hence, we set λ_{MMD} to $1e^5$. Given that our models are light, the training over instrument pairs or triplets can be done in less than 24 hours on a single mid-range GPU (eg. NVIDIA TITAN Xp 12Gb).

One-to-one transfer

First, we compare our MoVE proposal to UNIT on the one-to-one transfer task. To do so, we learn a different model for each pair of instruments. We perform incremental comparisons by ablating certain aspects of our proposal to assess their importance. First, we add concatenative conditioning of pitch and octave to UNIT (noted UNIT(GAN;C-po)). Then we add our proposed alternative MMD criterion replacing the GAN objective (UNIT(MMD;C-po)). Then, we introduce the FiLM layers leading to our MoVE proposal. The first version still features separate domain-specific encoders and decoders, so it is noted MoVE* (MMD; F-po). By further introducing domain conditioning and relying on a single VAE (Figure 27), we obtain our proposal MoVE (MMD; F-pod).

	reconstructions				**transfers**	
	RMSE	LSD	MMD ($\alpha = 0.05$)	k-NN (k = 10)	MMD ($\alpha = 0.05$)	k-NN (k = 10)
UNIT (GAN)	0.3412	718.47	2.117e-2	57269	2.038 e-2	43180
UNIT (GAN; C-po)	**0.3011**	693.22	1.989 e-2	57806	9.112 e-2	43414
UNIT (MMD; C-po)	0.3036	**692.41**	2.125 e-2	**57102**	2.304 e-2	43878
MoVE* (MMD; F-po)	0.3134	762.51	9.632 e-3	57273	3.153 e-2	43443
MoVE (MMD; F-pod)	0.3339	781.11	**2.587 e-3**	57509	**1.747 e-2**	**43173**

Evaluations of various models on the test sets.

Evaluation scores. To evaluate reconstruction performances, we compute several criteria between input samples x and reconstructions $\tilde{x}$. The Root-Mean-Square Error $RMSE = \sqrt{\sum(x-\tilde{x})^2}$ and Log-Spectral Distortion $LSD = \sqrt{\sum(10*\log_{10}(\frac{x^2}{\tilde{x}^2}))^2}$ provide different assessments of how various models are able to reconstruct samples from the test set. Therefore, they only assess reconstruction abilities without domain transfer. To evaluate the quality of domain transfers, we compute the Maximum Mean Discrepancy (MMD) and the non-differentiable k-Nearest Neighbour (k-NN) test [82]. Both are dissimilarity measures computed between the target data distribution and transferred samples. Hence, we evaluate test set transfers between different target domains.

Reconstruction and domain transfer. The averaged reconstruction and transfer results are presented in the following table, while separate evaluations for different pairs are in Annex B. As we can see, the UNIT-MMD model obtains the highest within-domain reconstruction score, while the MoVE model achieves better domain translation. Hence, it appears

that the MMD increases reconstruction performance, and that the FiLM conditioning ameliorates the transfer. It also seems that relying on a single encoder and decoder for domain transfer might provide better generalization, as can be verified by looking at the relative MMD and kNN scores on the transfer task. Indeed, it seems that the modulated but separate layers approach perform worse, while the single architecture performs better on most evaluations.

Audio descriptors topology. Audio descriptors are features used to compare the qualities of different sounds [203]. Hence, we rely on these to assess the effect of transfer, while providing a deeper understanding of its behavior. We compute the spectral *flatness*, *centroid*, *roll-off* and *loudness* on test samples reconstructed on their own domain or transferred to the other domain. Distribution and sample-specific plots for the spectral centroid are presented in Figure 28.

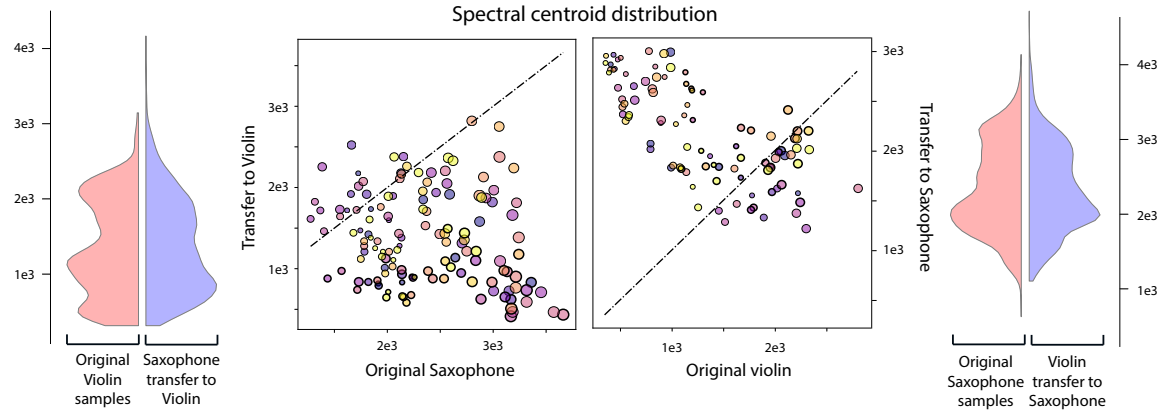

Figure 28: Understanding the effect of musical timbre transfer through audio descriptor distributions.

As we can see, the transfer produces an almost exact match of the descriptor distribution to the target domain. This shows the success in transferring multimodal distributions of auditory properties, as all the modes of the descriptors' distributions are preserved. The scatter plot also suggests that the centroid transfer is highly influenced by the loudness of the sample. This correlates to perception studies, as playing an instrument louder usually leads to a higher centroid [182].

In order to further understand how the latent space is organized with respect to audio descriptors, we provide their spatial topology in Figure 29. To compute this, we define a sampling grid over the latent space and decode the audio at each point to compute their descriptors. As we can see, the audio descriptors are locally very smooth. Furthermore, one

key observation is that the latent space of both conditioned target domains follow the same overall topology. Animations showing the complete latent descriptor topology are available on the supporting webpage.

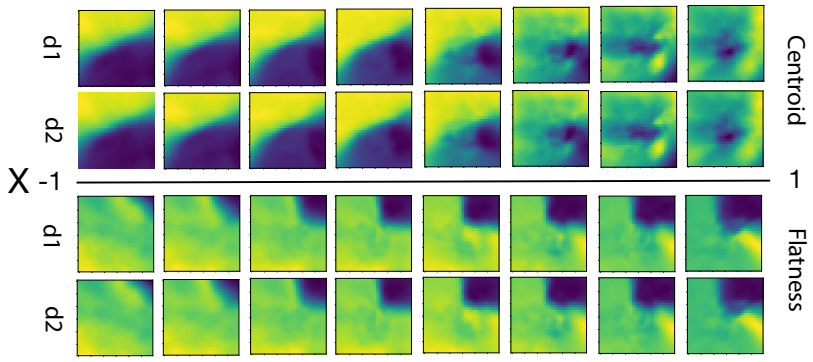

Figure 29: Topology of the latent space with respect to audio descriptors.

Latent space synthesis and performance. As the latent space provides continuous audio synthesis and that our method introduce high-level conditioned controls, we can use our proposal as a full musical synthesizer. Furthermore, as we map to 3-dimensional spaces, the user can directly interact with the space while performing timbre transfer. Furthermore, although these models are trained to transfer single instrumental notes, they still can be used to transfer a full melody recording between timbre domains. To do so, the recording is split and iteratively reconstructed by transfering each signal window to the target domain. Audio examples of applying this strategy to transfer a complete instrumental solo are also available in the supporting webpage.

Many-to-many transfer

Here, we evaluate the application of MoVE to perform many-to-many transfer. Given our new architecture, this simply consists in training on multiple domains at once by modulating with the appropriate domain information. This architecture allows us to train a single model for different domains and thus to perform multi-domain translation. The conditioning vector is then composed of the pitch, the octave and here the instrument of the corresponding example. This conditioning vector is then processed by an embedding to ease the FILM conditioning. Results are presented in the following table: we can see that the MoVE architecture is able to reconstruct and transfer multiple domains at the same time at the cost of a slight decrease

in performance, even in the case of diverse domains (here Alto-Saxophone, Flute, Violin and French-Horn).

	averaged reconstructions				**averaged transfers**	
	RMSE	LSD	MMD ($\alpha = 0.05$)	k-NN (k = 10)	MMD ($\alpha = 0.05$)	k-NN (k = 10)
Alto-Saxophone	0.5327	835.67	2.117 e-2	42299	2.386 e-2	59157
Flute	0.4593	761.46	2.119 e-2	49719	1.975 e-2	57277
Violin	0.3271	773.65	5.659 e-3	58013	1.452 e-2	55379
French-Horn	0.6239	869.69	3.404 e-3	70946	2.086 e-2	51317

Many-to-Many MoVE reconstruction & transfer scores.

Conclusion

We introduced the Modulated Variational auto-Encoders (MoVE), which perform many-to-many domain transfer within a single architecture and without adversarial training while providing high-level control over the generation. We effectively adapted this technique to musical timbre transfer and showed the successes of our method for audio synthesis. As our technique is generic, it could be applied to other types of data such as image or video. The architecture itself opens up a range of potential sonic applications such as playing style conditioning, transfers between acoustical and electronic instruments, and even with non-musical sound domains. Another avenue of research to be investigated is controlling the amount of transfer performed by the model.

5.2 Learning Representations of Timbre

Timbre representations by conditioning. Implicit timbre models such as those presented in the field of domain translation treat timbre as a categorical class represented by distinct datasets across which we learn a generative mapping. While this effectively assesses the task of unpaired conversion for input samples drawn from the source distribution, it does not offer explicit controls on the sampling process of timbre features besides the provided input data example and the categorical domain target. To this extent, we carried experiments in learning generative latent timbre representations which are invertible spaces of analysis and synthesis. One approach for controllable semantic manipulations was proposed with the Fader Networks [156] which comprise an auto-encoder and an adversarial latent regularizer to disentangle semantic controls in the learned representation. As suggested by the name, we train specific dimensions to account for specific attributes of the data which can be intuitively controlled as sliding faders while sampling the remaining latent dimensions. This is achieved by training a latent classifier at predicting attributes from the encoder output latent codes, while the encoder competes at fooling the classifier discriminator. In this setting, the regularised features cannot account for the attributes and the decoder must use specific conditioning dimensions to reconstruct semantically correct data.

We propose a musical note sampler with semantic controls over pitch, instrument timbre and playing style as fader dimensions, using the annotations available in the SOL database. We choose the Wasserstein Auto-Encoder (WAE) as baseline for learning a latent representation of Mel-scaled magnitude spectrograms, which uses the MMD as regularisation between the aggregated encoder posterior and the isotropic gaussian prior distribution from which we perform ancestral sampling. This alternative of the VAE with Kullback-Leibler divergence is expected to generate more accurate reconstructions by allowing a flexible aggregated regularisation rather than minimising the sample-wise divergence. Each note is down-sampled to 22050Hz, trimmed at the attack and transformed into log-magnitude spectrograms with 128 frames (about 1.6 second duration) and 500 Mel bins. The encoder output latent codes are adversarially trained at being invariant to the fader attributes which are provided as conditioning dimensions to the decoder via FiLM layers on its normalised hidden activations. To prevent checkerboard artefacts of transposed convolutions, we apply up-sampling in the decoder with nearest-neighbour interpolation followed by regular convolutions with unit stride. In the first

place, we perform audio inversion using the Griffin-Lim algorithm which prevents from fast rendering because of its iterative procedure. In the second place we pretrain a Multi-head CNN (MCNN [9]) which is a feed-forward neural vocoder that generates waveform in a single pass and use a larger database of musical notes as training data for the inversion model (subset of the Vienna Symphonic Library [12]). In this setting we can pipeline the output samples from the WAE decoder to the MCNN and generate audio in a single pass. However the model is not trained end-to-end to the waveform and reconstruction errors of the decoder may compound with those of the MCNN, a phenomenon which also happens with GLA approximate phase reconstruction.

In order to evaluate the accuracy of the conditioning on the generated output samples, we pretrain data classifiers as references to predict the attribute targets, either the note classes (semitone and octave) or the style classes (instrument or playing style). We sample the prior of conditional WAEs with random attribute targets, decode and rate by classification, a high average accuracy means that the generated samples corresponded to the correct semantic targets. We observe that the Fader regularisation leads to a more accurate conditioning by a large margin, although this is traded-off with a decreased spectrogram reconstruction accuracy compared to the WAE baseline and conditional WAEs without adversarial latent classification. We also measure the correlation of the learned representation with respect to attribute classes by MMD between latent encodings of each class. As expected the effect of the Fader regularisation tends to produce a latent space with lower separation between class distributions which in turns leads to a more effective conditioning. Some additional visualisations and audio samples are hosted on the dedicated online repository: `https://github.com/acids-ircam/Expressive_WAE_FADER`. This work was submitted to the 22$^{\text{nd}}$ International Conference on Digital Audio Effects (DAFx-19) and presented as an oral.

[12] `https://www.vsl.co.at/en`

Assisted Sound Sample Generation with Musical Conditioning in Adversarial Auto-Encoders

Adrien Bitton, Philippe Esling, Antoine Caillon & Martin Fouilleul

(original publishing template of DAFx 2019 available at
`https://arxiv.org/abs/1904.06215`)

Deep generative neural networks have thrived in the field of computer vision, enabling unprecedented intelligent image processes. Yet the results in audio remain less advanced and many applications are still to be investigated. Our project targets real-time sound synthesis from a reduced set of high-level parameters, including semantic controls that can be adapted to different sound libraries and specific tags. These generative variables should allow expressive modulations of target musical qualities and continuously mix into new styles.

To this extent we train auto-encoders on an orchestral database of individual note samples, along with their intrinsic attributes: note class, timbre domain (an instrument subset) and extended playing techniques. We condition the decoder for explicit control over the rendered note attributes and use latent adversarial training for learning expressive style parameters that can ultimately be mixed. We evaluate both generative performances and correlations of the attributes with the latent representation. Our ablation study demonstrates the effectiveness of the musical conditioning.

The proposed model generates individual notes as magnitude spectrograms from any probabilistic latent code samples (each latent point maps to a single note), with expressive control of orchestral timbres and playing styles. Its training data subsets can directly be visualized in the 3-dimensional latent representation. Waveform rendering can be done offline with the Griffin-Lim algorithm. In order to allow real-time interactions, we fine-tune the decoder with a pretrained magnitude spectrogram inversion network and embed the full waveform generation pipeline in a plugin. Moreover the encoder could be used to process new input samples, after manipulating their latent attribute representation, the decoder can generate sample variations as an audio effect would. Our solution remains rather light-weight and fast to train, it can directly be applied to other sound domains, including an user's libraries with custom sound tags that could be mapped to specific generative controls. As a result, it fosters creativity and intuitive audio style experimentations.

Introduction

Modern music production techniques rely on large and heterogeneous sound sample libraries along with diverse digital instruments and effects. It opens to a great variety of sound design possibilities and limitless contents to compose with, however principled interactions and scaled visualisations are still lacking in order to efficiently explore such potential and use it to generate *target sound qualities*.

Unsupervised generative models learn an underlying data distribution solely based on the observation of examples, in order to consistently generate novel content. They have been successfully applied to complex computer vision tasks such as processing facial expressions, landscapes, visual styles and paintings. Some solutions to audio emerged more recently, including pioneer musical systems such as *NSynth* (Neural Synthesizer [71]) for real-time high-quality sound synthesis. However, the heavy model architecture and prohibitive training time restrict its dissemination. The learned internal representation remains mostly uninformative and its many generative parameters are still too little correlated to explicit semantic qualities.

In this paper, we develop a high-level sound synthesis system with meaningful data visualisations and explicit musical controls. It is a lighter *non-autoregressive model* that can be trained fast on small datasets, including an user's personal libraries. Our goal is to learn expressive style variables from any sound tags, so that the model fosters creativity and *assists digital interactions* in music production. Considering note samples of orchestral instruments, we could for instance synthesise novel timbres or *playing style hybrids*.

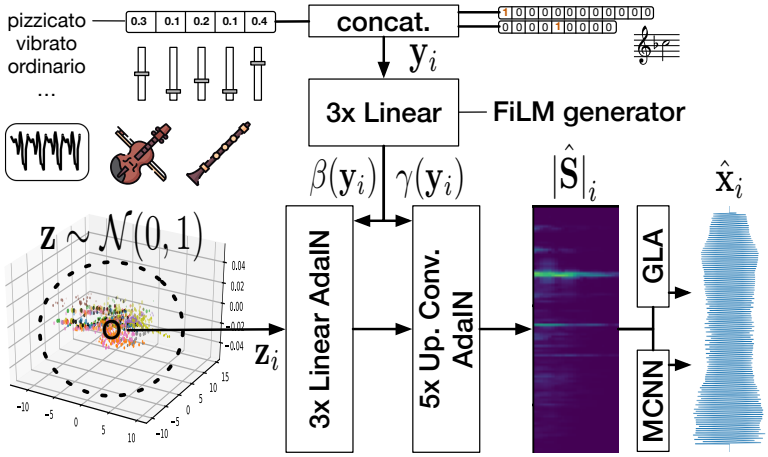

Figure 30: High-level note sample generation from the latent representation and musical conditioning in the decoder with FiLM. Intermediate features are modulated by the note targets and expressive style controls in order to synthesize new timbres and effects.

We train *Wasserstein Auto-Encoders* (WAEs [263][303]) on Mel- spectrogram magnitudes to

organise a generative latent representation of individual note samples spanning the *tessitura* of 12 orchestral instruments. The considered database has intrinsic attributes: note classes, playing styles and timbres (each instrument subset), that we wish to control when generating new notes from the latent space. Thus we extend the WAE model with musical conditioning in the decoder and *Adaptive Instance Normalization* (AdaIN [121]). Using *Feature Wise Linear Modulation* (FiLM [204]) and adversarial training with a *Fader latent discriminator* [156], our WAE-Fader model effectively learns these generative controls along with expressive style variables that can be mixed continuously.

We evaluate these features in terms of generative performances and representation. We perform an *ablation study* and show that the model can sustain a good test reconstruction quality while achieving an accurate attribute-conditional generation. The success of the method relies on an attribute-free latent representation so that the decoder is pushed to learn the conditioning. These distributions can be visualized directly in the 3-dimensional latent space where clusters denote an undesired attribute encoding. We measure it with inter-class statistics and latent post-classification. The experiment validates correlations between low attribute encoding and effective conditioning.

We obtain an expressive note sample generator with *3-dimensional* representations of the training sound domains, decoding *probabilistic latent samples* with explicit control over the rendered note qualities. The learned style variables of the orchestra can ultimately be mixed continuously, as *faders* do, in order to intuitively explore new musical effects. Generated spectrogram magnitudes can approximately be inverted to waveform with the *Griffin-Lim* iterative algorithm (GLA [100]). Ultimately we fine-tune the decoder with a pretrained inversion network [9] for *real-time waveform synthesis*. We embed the resulting generative system in a *plugin* allowing for *MIDI* mapping, live exploration and *Digital Audio Workstation* (DAW) integration.

State-of-art

Generative models and regularized auto-encoders. *Generative models* aim to find the underlying probability distribution of the data $p(\mathbf{x})$ based on a set of examples in $\mathbf{x} \in \mathbb{R}^{d_x}$. To do so, we consider *latent variables* defined in a lower-dimensional space $\mathbf{z} \in \mathbb{R}^{d_z}$ ($d_z \ll d_x$), a higher-level representation that could have led to generate any given example. The latent variable generative model is defined by the joint probability distribution $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, where the *prior* $p(\mathbf{z})$ is usually modelled with simpler distributions such as Gaussian or uniform while a complex conditional distribution $p(\mathbf{x}|\mathbf{z})$ maps latent codes to the data space. The model could be evaluated with the maximum marginal likelihood over

the considered dataset. However for complex distributions that could model real-world data, integration cannot be computed in closed form.
Regularized auto-encoders have been used to reformulate the problem as an optimization by jointly learning the generative mapping $p_\theta(\mathbf{x}|\mathbf{z}) \in \mathcal{G}$ and an encoding distribution $q_\phi(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}$ from families $\mathcal{G}, \mathcal{Q}$ of *approximate densities* both parameterized with neural networks. This was initially proposed through *Variational Inference* in the *Variational Auto-Encoder* (VAE [145]) that maximizes a lower bound of the data log-likelihood:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - D_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big] \leq \log p_\theta(\mathbf{x})$$

This amounts to optimizing the *Evidence Lower Bound Objective* (ELBO) that can be interpreted as follow, the first term is the Negative Log-Likelihood (NLL) data reconstruction cost and the second is the Kullback-Leibler Divergence (KLD) that quantifies the error made by using the approximate $q_\phi(\mathbf{z}|\mathbf{x})$ rather than the true $p_\theta(\mathbf{z})$. This latent regularization pushes the encoder to remain close to the prior latent density and can be weighted with a β parameter that balances these two objectives.

$$\mathcal{L}^{\text{ELBO}}_{\theta,\phi} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] + \beta \cdot D_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big]$$

The VAE is implemented with a *stochastic encoder* that parameterises an isotropic Gaussian latent distribution $q_\phi(\mathbf{z}|\mathbf{x})$
$\sim \mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x}))$ regularized against an unit variance prior. These assumptions allow analytical KLD computation and differentiable latent sampling for direct optimization of the ELBO.
The KLD forces each *individual* latent code to resemble the prior, which implicitly matches the whole encoded distribution. However a fitted ELBO value does not always result in an *effective inference*. Since the latent codes of different inputs are individually regularized, the KLD may prevent the encoder from learning any useful features (*posterior collapse* [302]) while the decoder only produces $p_\theta(\mathbf{x})$ regardless of the encoded information. Other conflicting solutions of the ELBO lead to undesired solutions and known limitations of VAEs such as *blurriness* of generated samples or *uninformative latent dimensions* ([296]).
With justifications stemming both from *Likelihood-free Optimization* (InfoVAE [303]) and the theory of *Optimal Transport* (WAE [263]), a more general framework for training regularized auto-encoders was recently proposed and that we call Wasserstein Auto-Encoders (WAEs). Considering a *deterministic decoder* $G_\theta : \mathbf{z} \to \mathbf{x}$ and any family of conditional encoder distribution $Q_\phi(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}$, it is sufficient that the marginal $Q_Z(\mathbf{z}) := \mathbb{E}_X\big[Q(\mathbf{z}|\mathbf{x})\big]$ matches any prior P_Z. In comparisons with VAEs, WAEs can optimize any non-negative cost function C and *any divergence measure* D_Z between latent distributions, without requiring a stochastic

encoder nor restricting the latent model to Gaussian prior:

$$\mathcal{L}_{\text{WAE}} := \inf_{Q(\mathbf{z}|\mathbf{x})\in\mathcal{Q}} \mathbb{E}_X \mathbb{E}_{Q(\mathbf{z}|\mathbf{x})}\big[C(\mathbf{x}, G(\mathbf{z}))\big] + \beta \cdot D_Z(Q_Z(\mathbf{z}), P_Z)$$

Thus we set our experiment in the more flexible WAE framework. These regularized auto-encoders are powerful unsupervised representation learning models, rather light-weight and fast to train, performing both inference (encoder) and generation (decoder). They are effective on small datasets (hundreds of training examples), learning a structured latent representation with disentangling capacities encouraged when $\beta > 1$. Once trained, probabilistic samples of the latent prior are consistently decoded into new samples and latent interpolations map to smooth data variations.

Maximum Mean Discrepancy Regularization. As shown for VAEs, the choice of *latent divergence* heavily impacts the resulting model performances. Since the point-wise KLD has strong intrinsic limitations, a more flexible regularization is required for WAEs. Such *differentiable* divergence on latent distributions was developed in the *Reproducing Kernel Hilbert Space* (RKHS) as a distance between probabilistic moments $\mu_{p,q}$ computed with a non-parametric kernel k:

$$||\mu_p - \mu_q||_H^2 = \langle \mu_p - \mu_q, \mu_p - \mu_q \rangle = \mathbb{E}_{p,p} k(x, x') + \mathbb{E}_{q,q} k(y, y') - 2\mathbb{E}_{p,q} k(x, y)$$

It defines the *Maximum Mean Discrepancy* (MMD [97]) between two distributions $x \sim p(x)$ and $y \sim q(y)$, where $\mathbb{E}_{p,q}$ is the expectation that can be evaluated with the *Radial Basic Function* (RBF) kernel of free parameter Σ:

$$k_{(\text{RBF})}(x, y) = \exp\left(\frac{||x - y||^2}{-2\Sigma^2}\right)$$

To the extent of latent regularization, MMD can be computed between every deterministic mini-batch encoding $\mathbf{z}_{\text{encoder}} = Q(\mathbf{x})$ and random samples from any latent prior $\mathbf{z}_{\text{prior}} \sim P_Z$. Throughout the model optimization, MMD is thus matching the aggregated encoder posterior to the prior rather than regularizing each latent point individually. In comparisons with KLD, WAE-MMD allows for less constrained inference and richer latent representations. For instance, increasing β to two orders of magnitude above the reconstruction cost does not impede the decoder training. Since the WAE objective does not optimize the bounded NLL, the overall generative performances can be improved.
Other kernel functions can be used, which may be more discriminating at the expense of heavier computations. Alternatively to MMD, the WAE-GAN uses an adversarial latent

discriminator to assess the divergence, thus optimizing a parametric function that could match even closer the encoder to the prior. However, since we consider a low-dimensional latent space of only 3 dimensions and remain with a simple isotropic Gaussian prior, MMD-RBF is sufficient yet light and stable to train on.

Conditioning and feature normalizations. Regularization in auto-encoders encourages disentanglement of independent generative factors onto separate latent dimensions that would in turn control the corresponding decoded data variations. However this is only partly achieved on toy datasets (β-VAE [111]) and in most cases the unsupervised latent dimensions are hardly related to explicit generative parameters. An additional supervision signal may be applied to the generative neural network in order to control and render specific attributes of the data. Thus we consider observations $\mathbf{x}$ paired with attribute annotations $\mathbf{y}$, and condition the decoder as $G : \{\mathbf{z}, \mathbf{y}\} \rightarrow \mathbf{x}$.

The simplest conditioning for categorical attributes is to encode them into one-hot vectors that are concatenated to the latent codes before being processed by the decoder. However more advanced conditioning techniques have been developed as for visual style transfer, using full images as conditions (conditional style transfer [90]). In the *Feature-wise Linear Modulation* (FiLM [204]) approach, a separate generator learns a mapping from any style inputs to adaptive biases $\beta_{\text{FiLM}}(\mathbf{y})$ and scales $\gamma_{\text{FiLM}}(\mathbf{y})$ applied to the conditional network computations. This modulation may be placed anywhere within the architecture and proved to be particularly suited to *Adaptive Instance Normalization* (AdaIN [121]). Considering the *l-th* hidden layer output activations $\mathbf{h}^l = g^l(\mathbf{h}^{l-1})$ of a generative neural network, the conditional modulation is thus be computed as:

$$\text{AdaIN}(\mathbf{h}^l, \mathbf{y}) = \gamma^l_{\text{FiLM}}(\mathbf{y}) \left[\frac{\mathbf{h}^l - \mu(\mathbf{h}^l)}{\sigma(\mathbf{h}^l)} \right] + \beta^l_{\text{FiLM}}(\mathbf{y})$$

in which mean and standard deviation $\{\mu,\sigma\}$ are computed across features, independently for each channel and each sample. In the context of style transfer, it can be interpreted as aligning the mean and variance of the content features with those of the style condition. It is a versatile conditioning technique, requiring little additional computations (particularly when applied channel-wise in convolution layers). It also suits well to handling multiple conditions that may more efficiently be mapped throughout the network rather than arbitrarily concatenated to the input. Thus we will use FiLM and AdaIN for conditioning the decoder on both note and style classes. However, such normalization is not suited to classification tasks since content features are individually normalized. In order to preserve its inference power, we will use *Batch Normalization* (BN) on the encoder's hidden activations.

Adversarial latent training. Adversarial regularization was proposed as an alternative to MMD in the WAE-GAN. For simple low-dimensional latent distributions, the expense of an additional parametric adversarial regularizer is not required. Nonetheless, adversarial latent training remains relevant for expressive conditioning. As detailed in the previous section, adaptive conditioning techniques paired with specific feature normalizations substantially improved feed-forward style transfer. However, in an auto-encoder setting, if the latent space implicitly encodes the attributes of interest, the decoder bypasses the conditioning and does not learn any effective generative controls. This problem was tackled in image generation with the introduction of an adversarial *Fader* latent discriminator $\mathcal{F}$ (Fader Networks [156]) that competes with the non-conditional encoder in order to prevent correlations between attributes and latent distributions. As for the conditional models, we consider annotated data samples $\{\mathbf{x}, \mathbf{y}\}$ and for simplicity, a categorical one-hot representation $\mathbf{y} \in \{0, 1\}^n$ with a single $y_i = 1$ and its opposite $\bar{\mathbf{y}} := \mathbf{1}^n - \mathbf{y}$. Such attribute-free latent representation is implemented in two separate optimization steps, first latent classification of the true attribute $\mathcal{F} : \mathbf{z} \to \hat{\mathbf{y}} \sim p_\psi(\mathbf{y}|Q(\mathbf{x}))$, then adversarial confusion of the latent classifier at predicting the opposite:

$$\mathcal{L}^{\text{class.}}(\psi|\phi) = -\sum_{\mathbf{x},\mathbf{y}} \log(p_\psi(\mathbf{y}|Q(\mathbf{x})))$$
$$\mathcal{L}^{\text{adv.}}(\phi|\psi) = -\sum_{\mathbf{x},\mathbf{y}} \log(p_\psi(\bar{\mathbf{y}}|Q(\mathbf{x})))$$

As the encoder is pushed to remain invariant to attributes, the decoder is forced to learn the conditioning in order to reconstruct every input samples along with their source attributes. Thus it replaces adversarial training in the high-dimensional pixel space with latent attribute confusion in the low-dimensional latent space in order to efficiently learn style transfer variables. Applied to facial expressions, these *Fader* variables can continuously modulate complex visual features such as gender (female $\leftrightarrow$ male) or age (younger $\leftrightarrow$ older). Moreover, in mixing several attributes, one could generate new style qualities.

Audio synthesis. Neural networks can be trained on spectrogram magnitudes (and other spectral features) for audio analysis purpose. It eases the subsequent modelling task, often involving pattern detection, from a pre-processed structured sound representation. However, for generative purpose, an inversion from magnitude to waveform is required since the complex phase information was discarded. It is commonly done offline with GLA [100]. Further advances in generative neural networks for audio have targeted raw waveform modelling through specific architecture design. *Wavenet* [268][71] is amongst them the most popular solution. It uses several stacks of dilated causal convolutions in order to aggregate multiple temporal

granularities and structure long-term dependencies, which is challenging at the high audio sample rate. The output is a single auto-regressive sample prediction given all the previous sample context $p(x_t|x_1..x_{t-1})$. It results in high-quality real-time audio synthesis. However this sample level modelling requires long training times, heavy architectures that offer little knowledge over their learned features.

The *Multi-head Convolutional Neural Network* (MCNN [9]), a recent alternative for audio waveform modelling, was designed as a feed-forward real-time magnitude spectrogram inversion system that is not restricted to linear frequency scale. It proved to outperform GLA quality for speech. The use of differentiable GPU-based STFT computations enables a faster optimization onto spectral losses, rather than *auto-regressive* sample predictions:

$$\mathbf{x} \xrightarrow{|\text{STFT}|} |\mathbf{S}| \xrightarrow{\text{MCNN}} \hat{\mathbf{x}} \xrightarrow{\text{STFT}} \hat{\mathbf{S}} \Longrightarrow \mathcal{L}^{\text{MCNN}}(\mathbf{S}, \hat{\mathbf{S}})$$
$$\mathcal{L}^{\text{MCNN}} = \lambda_0 \cdot \mathcal{L}^{\text{SC}} + \lambda_1 \cdot \mathcal{L}^{\text{logSC}} + \lambda_2 \cdot \mathcal{L}^{\text{IF}} + \lambda_3 \cdot \mathcal{L}^{\text{WP}}$$

where $|\mathbf{S}|$ can be any spectrogram magnitude (including Mel-scaled frequencies). The model is well tailored to audio with multiple heads of 1-dimensional temporal up-sample convolutions. These heads focus on different spectral components and sum into waveform. It remains light-weight and could be adapted in an end-to-end waveform auto-encoder. Four objectives were originally proposed, using the complex STFT for the *Instantaneous Frequency* (IF) and *Weighted Phase* (WP) losses, that we could not optimize successfully. Hence we will only use the *Spectral Convergence* (SC) and log-scale magnitude (logSC) losses:

$$\mathcal{L}^{\text{SC}}(\mathbf{S}, \hat{\mathbf{S}}) = \| |\mathbf{S}| - |\hat{\mathbf{S}}| \|_F / \| |\mathbf{S}| \|_F \text{ with } \|.\|_F \text{ the Frobenius norm}$$

$$\mathcal{L}^{\text{logSC}}(\mathbf{S}, \hat{\mathbf{S}}) = \| \log(|\mathbf{S}| + \epsilon) - \log(|\hat{\mathbf{S}}| + \epsilon) \|_1$$

Method

Our experiment begins with the WAE-MMD, isotropic unit variance Gaussian prior $\mathbf{z}_{\text{prior}} \sim \mathcal{N}(0, 1)$, RBF kernel and BN in both encoder and decoder in order to structure a 3-dimensional generative latent sound representation. Given a magnitude spectrogram $|\mathbf{S}|$ and a corresponding set of annotated attributes $\mathbf{y}$, we are learning $Q : |\mathbf{S}| \rightarrow \mathbf{z}$ and $G : \mathbf{z} \rightarrow |\hat{\mathbf{S}}|$ such as $|\mathbf{S}| \approx |\hat{\mathbf{S}}| = G(Q(|\mathbf{S}|))$ with *Binary Cross-Entropy* (BCE) reconstruction cost:

$$\mathcal{L}_{\text{WAE}} = \text{BCE}(|\mathbf{S}|, |\hat{\mathbf{S}}|) + \beta \cdot \text{MMD}_{\text{RBF}}(\mathbf{z}, \mathbf{z}_{\text{prior}})$$
$$\text{BCE}(x, \hat{x}) = -\big[x \log \hat{x} + (1 - x) \log(1 - \hat{x})\big] \; ; \; |x| < 1$$

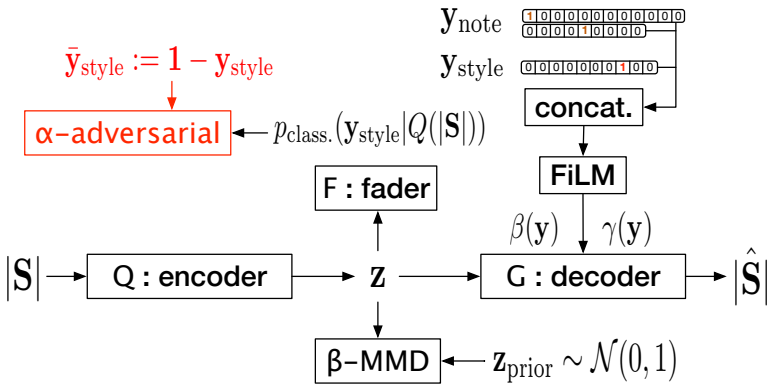

Figure 31: How information flows in the adversarial optimization of the WAE-Fader

We can sample random codes from the latent prior and consistently decode new magnitude samples, however there is no control on the output features. For the orchestra, we consider $\mathbf{y} = \{\mathbf{y}_{\text{note}}, \mathbf{y}_{\text{style}}\}$ with $\mathbf{y}_{\text{note}} = \{\text{semitone, octave}\}$. We define the timbre attribute as the class of an instrument subset, which comprises the *Ordinario* mode as well as diverse extended playing techniques such as *Staccato*, *Flatterzunge* or *Pizzicato*. When considering a single subset, we thus aim at controlling the playing techniques of the considered instrument as $\mathbf{y}_{\text{style}}$. When considering multiple instruments, instead we aim at controlling the different timbres, either in *Ordinario* or with mixed playing styles within each instrument subset.
For explicit controls over the rendered attributes, we condition the decoder as $G : \{\mathbf{z}, \mathbf{y}\} \rightarrow |\hat{\mathbf{S}}|$ using AdaIN. An additional FiLM generator is fed with concatenated one-hot vectors of the three attribute classes (semitone, octave and style). It learns an adaptive mapping to biases $\beta_{\text{FiLM}}(\mathbf{y})$ and scales $\gamma_{\text{FiLM}}(\mathbf{y})$ that are used to modulate the normalized decoder activations. In order to effectively learn the style conditioning and expressively modulate timbres or playing techniques, we use adversarial training with a Fader latent discriminator $\mathcal{F} : \mathbf{z} \rightarrow \hat{\mathbf{y}}_{\text{style}} \sim p_{\text{class.}}(\mathbf{y}_{\text{style}}|Q(|\mathbf{S}|))$ that competes with the non-conditional encoder in classifying the considered styles from latent codes:

$$\mathcal{L}_{\text{class.}} = -\log p_{\text{class.}}(\mathbf{y}_{\text{style}}|Q(|\mathbf{S}|))$$
$$\mathcal{L}_{\text{WAE-Fader}} = \mathcal{L}_{\text{WAE}} - \alpha \cdot \log p_{\text{class.}}(\bar{\mathbf{y}}_{\text{style}}|Q(|\mathbf{S}|))$$

with $\bar{\mathbf{y}}_{\text{style}} := \mathbf{1} - \mathbf{y}_{\text{style}}$ and α that weights the adversarial loss in the encoder. Classification is optimized on the NLL with *Softmax* probilities. The resulting attribute confusion prevents the latent space from implicitly encoding the style distributions, thus the decoder is forced to use the conditioning to reconstruct the source features from the attribute-free code. Ultimately these learned style variables can continuously be mixed, as actual *faders* do. We refer to this

final model as WAE-Fader, that still uses MMD regularization. It allows for controlling the strength of each rendered attribute and intuitively exploring hybrid sound effects from any custom tags, here either chosen from extended playing techniques or from diverse orchestral timbre domains.

The resulting generative system maps any latent coordinate $\mathbf{z} \sim \mathcal{N}(0,1) \in \mathbb{R}^3$ to target note spectrograms with expressive musical style controls. Inversion from spectrogram magnitudes to audio waveforms can be done offline with GLA. Alternatively, we pretrain a MCNN on a larger corpus of musical note samples to allow real-time rendering. In order to improve the final audio quality, we fine-tune the full generative model by freezing the encoder parameters and jointly optimizing the learned decoder with the pretrained MCNN as:

$$\mathbf{x} \xrightarrow{\text{STFT}} \mathbf{S} \xrightarrow{|\text{Mels}|} |\mathbf{S}| \xrightarrow{Q} \mathbf{z} \xrightarrow{G \circ \text{MCNN}} \hat{\mathbf{x}} \xrightarrow{\text{STFT}} \hat{\mathbf{S}} \Longrightarrow \mathcal{L}^{\text{MCNN}}(\mathbf{S}, \hat{\mathbf{S}})$$

This waveform pipeline $\{G \circ \text{MCNN}\}$ is embed in a plugin for live interactions and DAW integration. Using a MIDI interface, we can for instance trigger target note classes $\mathbf{y}_{\text{note}}$ with keys and map the continuous generative parameters to faders. These are the latent dimensions $\mathbf{z}$, that can also be randomly sampled, and most interestingly the adversarially learned style variables $\mathbf{y}_{\text{style}}$ that can be mixed to explore new sound effects.

Experiment

Dataset. We use the Studio-On-Line (SOL [12]) library of around 15000 individual note samples, across the tessitura of 12 orchestral instruments grouped in 4 families and with many extended playing techniques, that may be specific or shared across instrument families. These are *Wind* (Alto-Saxophone, Bassoon, Clarinet, Flute, Oboe), *Brass* (English-Horn, French-Horn, Tenor-Trombone, Trumpet), *String* (Cello, Violin) and *Keyboard* (Piano). Notes are consistently tagged with the intrinsic attributes of the dataset: note classes (12 semitones across 9 octaves), several dynamics and playing styles of every instrument. We define two style experiments for the orchestra. If training on a single instrument, we aim for expressive synthesis of its playing styles. If training on multiple instruments, we aim for timbre control. Each instrument subset defines a timbre domain, either in *Ordinario* (its common mode) or with all styles mixed.

Audio files are down-sampled to 22050Hz and pre-processed into Mel-spectrograms with a FFT size of 2048, hop size of 256 and 500 bins ranging the full spectrum. As we consider a generator of *individual* notes, we set a common audio length of 34560 samples ($\sim$1.6s) from the attack which amounts to 128 STFT frames. We choose this duration as a trade-off between input and latent dimensionality, limiting the amount of silence after shorter playing

modes (eg. *Pizzicato*) while keeping some sustain for longer notes (from which some sustain and decay may have been cropped). Magnitudes are floored to 1e-3 and log-scaled in [0,1] according to the BCE range. Each playing style subset of each instrument is split into 80% training, 10% validation and 10% test notes. In average each instrument has 10 playing styles and 100 to 200 notes for each.

Implementation details. Architecture of the WAE-Fader: Our experiments have been implemented in the *PyTorch* environment and our codes will be shared with this dependency. All convolution layers use 2-d. square kernels, an input zero-padding of half the kernel size and are followed by 2-dimensional feature normalization. All fully-connected linear layers are followed by 1-dimensional feature normalization. The non-linear activation used after every normalization is CELU. The deterministic encoder has 5 convolution layers with [12, 24, 48, 96, 128] output channels, kernel size 5 and stride 2, that down-sample the input spectrograms into 128 output maps that are flattened into an intermediate feature vector of size 8192. It is followed with a bottleneck of 3 linear layers of output sizes [1024, 512, 3] mapping to the latent space. For input Mel-spectrograms of size (500,128), it amounts to a dimensionality reduction of more that 5 orders of magnitude. All normalizations are BN. The decoder mirrors this structure with 3 linear layers of output sizes [512, 1024, 8192]. This vector is then reshaped into 128 maps. To avoid the known *checkerboard artifacts* [193] of the transposed convolution, we use *nearest neighbor* up-sampling followed with convolution of stride 1. These maps are processed with 4 up-sampling of ratios 3, the last one directly mapping to the input dimensionality of (500,128), and 5 convolutions with [96, 48, 24, 12, 1] output channels and kernel sizes [5, 5, 7, 9, 7]. All normalizations are AdaIN and the decoder output activation is sigmoid, bounded in [0,1] according to the BCE range. The FiLM conditioning is applied feature-wise at the output of the first two linear layers and channel-wise after. It amounts to 3688 modulation weights computed by an additional FiLM generator of 3 linear layers of output sizes [512, 1024, 3688]. Its output is split into biases and scales of sizes [512, 1024, 128, 96, 48, 24, 12]. The Fader latent discriminator has 3 linear layers of output sizes [1024, 1024, n_{style}] with *LeakyReLU* activations and a dropout ratio of 0.3, mapping latent codes to probabilities of the n_{style} classes.

Training parameters: We train our models with the Adam optimizer, an initial learning rate of 5e-4 and a batch size of 90. All model weights are initialized with Xavier uniform distribution. Depending on the considered data subset size, between 200 and 800 epochs are needed. A single instrument (1000-1500 notes) can be modelled in less than 2 hours on one NVIDIA TITAN Xp GPU. Training over all instruments and styles at once (around 11000

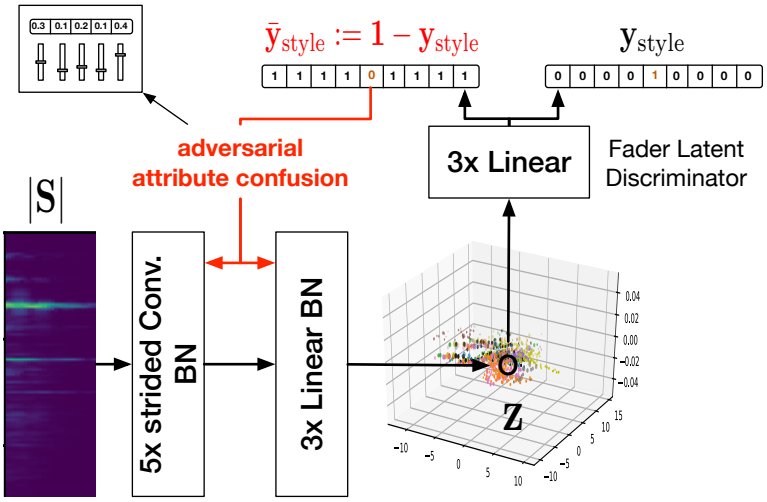

Figure 32: The Fader latent discriminator tries to infer the true style attribute while the encoder adversarially aims at fooling it. It encourages attribute invariance in the latent representation and learning of continuous generative controls in the decoder.

notes) takes less than 12 hours. In the first part of the training (30 to 100 epochs), we only optimize the reconstruction and classification objectives. Then we gradually introduce the MMD regularization (β-warmup) and the adversarial feedback in the encoder (α-warmup) until the first half of training epochs. The rest of the training jointly optimizes all training objectives at their target strengths $\beta = 40$ and $\alpha = 4$. These value were estimated in order to approximately balance the gradient magnitudes back-propagated by each loss. However, for the baseline WAE-MMD models we warmup β to 500 since it does not prevent from optimizing the reconstruction cost.

Signal reconstruction: The above described model trains on inputs with 128 frames of Mel-spectrogram, which amount to 34560 waveform samples according to our STFT settings. The generated Mel-magnitudes can be approximated back to the linear frequency scale and iteratively inverted with GLA for 100 to 300 iterations. To allow real-time rendering and a possibly improved audio quality, we reproduce the original MCNN architecture for Mel-spectrogram magnitudes inversion. We use 8 heads, $\lambda_0 = 1$ and $\lambda_1 = 6$. We could not successfully optimize the complex losses, however, we compute these magnitude losses on both the linear and Mel frequency scales. We pretrain this model on a larger dataset of around 50 hours audio comprising SOL and subsets of the *Vienna Symphonic Library* (VSL). Ultimately, we fine-tune the trained decoder with this pretrained MCNN. To do so, we freeze the encoder weights and optimize $G \circ \text{MCNN}$ on the model train set. The auto-encoder pair G, Q maps to Mel-spectrogram magnitudes $|\mathbf{S}|$ which are inverted to signals by the MCNN. However, the

loss computation $\mathcal{L}^{\text{MCNN}}(\mathbf{S}, \hat{\mathbf{S}})$ is not necessarily restricted to this frequency scale. Thus we evaluate and sum $\mathcal{L}^{\text{SC}}$ $\mathcal{L}^{\text{logSC}}$ on both linear and Mel frequency scaled magnitudes.

classified attribute (n_{style})	train set	validation	test
Semitone (12)	1.00	0.99	0.99
Octave (9)	1.00	0.99	1.00
Ordinario timbres (12)	1.00	1.00	1.00
Extended timbres (12)	1.00	1.00	1.00
Violin playing styles (10)	1.00	0.97	0.95
Clarinet playing styles (10)	1.00	0.96	0.94
Piano playing styles (10)	1.00	0.92	0.95
Trumpet playing styles (10)	1.00	0.92	0.94
Alto-Saxophone pl. styles (10)	1.00	0.98	1.00
Tenor-Trombone pl. styles (11)	1.00	0.90	0.90

*Reference **F1-scores** of the pretrained data classifiers used for the evaluation of conditional note generations*

Evaluations. **Generative performances:** First, we evaluate the ability of our models to produce accurate spectrograms by computing the reconstruction scores on the test set with *Root-Mean Squared Error* (RMSE) and *Log-Spectral Distance* $\text{LSD} = \sqrt{\sum \left[10 \log_{10}(|\mathbf{S}|/|\hat{\mathbf{S}}|)\right]^2}$. Regarding the conditioning aspects, we first pretrain data classifiers to reliably discriminate the different attribute classes and report their performances. These classifiers share the same architecture as the encoder but map to the n_{style} classes of interest. We use them as references to evaluate the effectiveness of the conditioning. Then, we sample an evaluation batch of 1000 random latent points from the prior, along with random semitone and octave targets. This evaluation batch is decoded to each attribute of the model (either playing styles or timbres) and classified with the corresponding reference classifier. A high accuracy means an effective conditioning for the task of musical note generation. We report the average accuracy for all the target conditions, with random octaves both in [0-8] (full orchestral range) or in [3-4] where models train on the overlap of every instrument tessitura.

Latent space structure: The effectiveness of the conditioning relies on learning an attribute-free latent representation of the data. If the attribute distributions are clustered, the decoder may learn their correlations with latent dimensions and bypass the conditioning signal. This phenomenon is alleviated with adversarial training of the non-conditional encoder against a Fader latent discriminator. As we map to 3-dimensional spaces, we can directly visualize this latent organization. We also propose two evaluations of the attribute representations. First,

model	test rec.		note cond. acc.				style cond. acc.	
	MSE	LSD	$st._{34}$	$oct._{34}$	$st._{08}$	$oct._{08}$	$style_{34}$	$style_{08}$
Violin playing styles (n_{style}=10) 1475 training note samples								
WAE-MMD	0.76	68.2	NA	NA	NA	NA	NA	NA
WAE-note	0.69	55.4	0.73	0.72	0.47	0.43	NA	NA
WAE-style	0.74	59.6	0.47	0.39	0.30	0.22	0.20	0.17
WAE-Fader	0.80	91.1	**0.96**	**0.77**	0.97	0.48	**0.88**	**0.93**
Ordinario timbres (n_{style}=12) 1784 training note samples								
WAE-MMD	1.04	88.3	NA	NA	NA	NA	NA	NA
WAE-note	0.84	71.6	0.99	0.96	0.62	0.53	NA	NA
WAE-style	0.80	65.7	0.64	0.58	0.30	0.24	0.33	0.19
WAE-Fader	1.01	105	**1.00**	**1.00**	0.94	0.68	**0.95**	**0.70**
Extended timbres (n_{style}=12) ¿11000 training note samples								
WAE-MMD	0.93	175	NA	NA	NA	NA	NA	NA
WAE-note	0.69	173	0.99	0.98	0.72	0.64	NA	NA
WAE-style	0.65	172	0.84	0.83	0.44	0.39	0.61	0.34
WAE-Fader	1.32	182	**1.00**	**1.00**	0.90	0.71	**0.95**	**0.64**

*The ablation study confirms the effectiveness of the **WAE-Fader** conditioning, both on target notes and playing styles or timbres. The conditional latent sampling is either performed with random octaves in [3,4] (the overlap of every tessitura) and [0-8] (the full orchestra range), we report the accuracy of the conditioning with respect to the targets $note_{34,08}$ (st. is semitone classe and oct. is octave classe) and $style_{34,08}$.*

we compute the average inter-class latent statistics with MMD. In this case, low values mean that the attribute distributions blend in the final representation. Second, we also perform a post-classification task by training classifiers at predicting the attributes from the learned latent representation. These models use the same architecture as the Fader discriminator, and we report their final accuracy. In this case, low scores mean that the latent representation did not encode the attributes.

Results

Ablation study. We defined both generative and representation evaluations to assess the effectiveness of our proposed musical conditioning. To study the benefits and compromises of each model feature, we train the base WAE-MMD and compare it with ablations of the WAE-Fader. The incremental model comparisons are WAE-MMD (no conditioning), WAE-note (semitone and octave conditioning), WAE-style (note and style conditioning) and

WAE-Fader. In order to simplify the notation, we do not specify the MMD but this regularization is used for all models. We performed this ablation study on the violin subset that has the following annotated playing styles: *Ordinario, Sustained, Short, Non-vibrato, Staccato, Pizzicato-secco, Medium-vibrato-short, Tremolo, Medium-vibrato-sustained* and *Pizzicato-l-vib*. We also compare the WAE-Fader on instrument timbres, either in ordinario or for all extended techniques mixed per instrument subset. It confirms the effectiveness of the expressive conditioning when the attribute-invariance assumption is achieved.
As we can see, conditioning WAE-note on the semitone and octave classes shows that the WAE-MMD model can partly learn the note controls with FiLM conditioning. Accordingly, the latent space structure does not exhibit strong correlations with the note classes anymore but with the style attributes that become the main unsupervised data feature. We also notice that this additional supervision improves the reconstruction quality. However, when adding the style conditioning in WAE-style, it seems that most performances drop. Indeed, the overall conditioning becomes little effective, both for the target note and style conditions. The final results show that the adversarial latent training enables the WAE-Fader model to effectively learn the complete conditioning, at the expense of a possible drop in its reconstruction accuracy.
It also seems that the task of modelling the playing styles when learning on a single instrument is more challenging than changing the timbres across multiple instruments. This can be seen in the lower performance of the WAE-style model applied to the violin. This may also be explained by the reduced size of the training data when the learning is restricted to single instrument subsets. These observations are supported by the resulting audio outputs of the conditional note generations. Indeed, it appears that meaningful and expressive variations when switching to any attribute conditions are only achieved with our proposed WAE-Fader model. This is successful for conditioning applied on both timbre attributes or playing styles.

Expressive note sample generations. In this section, we report additional experiments on the WAE-Fader models when conditioned on the playing styles of different instruments and families. Our model seems to train successfully on playing styles in every instrument families, as well as across the 12 instrument timbres of the orchestra as shown in the previous ablation study. This amounts to a great variety of sound qualities spanning extended modes of the orchestra, and let us hypothesize that the model could be applied to other sound domains as long as the tags are consistent with the data. Furthermore, the style variables learned with the Fader latent discriminator are continuous independent controls that can be mixed. Hence, this can allow our system to modulate the strength of rendered

model	inter-class MMD			post-class. acc.		
	st.	oct.	style	st.	oct.	style
Violin playing styles (n_{style}=10) 1475 training note samples						
WAE-MMD	0.25	0.26	0.30	0.92	0.94	0.56
WAE-note	0.03	0.10	0.50	0.04	0.49	0.82
WAE-style	0.12	0.16	0.35	0.25	0.55	0.59
WAE-Fader	0.02	0.01	0.46	0.08	0.34	0.64
Ordinario timbres (n_{style}=12) 1784 training note samples						
WAE-MMD	0.12	0.38	0.28	0.75	0.87	0.59
WAE-note	0.02	0.28	0.51	0.33	0.57	0.83
WAE-style	0.04	0.40	0.35	0.13	0.60	0.63
WAE-Fader	0.33	0.08	0.03	0.17	0.25	0.23
Extended timbres (n_{style}=12) ¿11000 training note samples						
WAE-MMD	0.02	0.41	0.12	0.71	0.89	0.48
WAE-note	5e-3	0.30	0.22	0.07	0.49	0.71
WAE-style	4e-3	0.32	0.18	0.07	0.56	0.55
WAE-Fader	3e-3	0.20	0.11	0.11	0.46	0.43

The ablation study allows to monitor the latent organization in the different models and throughout their training. We use both inter-class statistics and latent post-classification to estimate the final attribute invariance in the learned representation.

styles and create new effects by combining multiple attributes. Our model can also be used for sample modifications, akin to traditional audio effects, by encoding a given sample and manipulating the attribute conditions in order to decode different sample transformations.

Audio outputs and plugin development. As discussed previously, our proposed models can generate magnitude spectrograms, while controling their expressive qualities. These spectrograms can be either inverted to waveform offline with GLA or real-time if paired with MCNN. When fine-tuning the learned decoders with the pretrained MCNN on magnitude losses, we obtain a quality almost equivalent to the GLA approximation. We provide audio examples of test set reconstructions and conditional note generations inverted with both GLA and MCNN for individual listening evaluation on the companion webpage. While the audio quality of these results can still be improved, we can already confirm the ability of the model to provide semantic controls. As the learned style variables of WAE-Fader can be mixed continuously, we also provide some sound examples that were generated when modifying multiple orchestral attributes.

Our proposal provides intuitive sound synthesis of target sound qualities with learned style

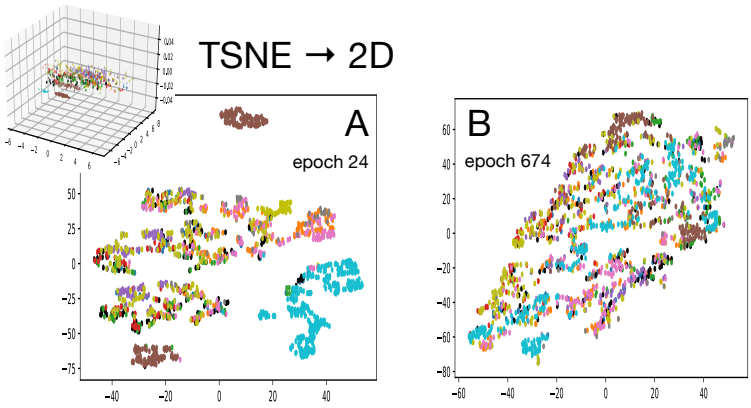

Figure 33: Latent organization as the WAE-Fader model trains on the ordinario timbres, each instrument domain being represented by a separate color. In ***A****, at epoch 24, the encoder does not optimize the adversarial loss yet. Its unsupervised representation exhibits the attribute classes. In* ***B****, at epoch 674, the α-warmup is finished and the adversarial latent training had blended the attribute distributions. The 2-dimensional projections are performed with t-Distributed Stochastic Neighbor Embedding (TSNE).*

model	test rec.		note cond. acc.				style cond. acc.		inter-class MMD			post-class. acc.		
	MSE	LSD	$st._{34}$	$oct._{34}$	$st._{08}$	$oct._{08}$	$style_{34}$	$style_{08}$	st.	oct.	style	st.	oct.	style
Clarinet	0.87	116	0.96	0.99	0.98	0.45	0.97	0.92	0.05	0.58	0.12	0.15	0.76	0.41
Piano	0.99	113	0.53	0.91	0.47	0.72	0.72	0.64	0.03	0.03	0.08	0.16	0.20	0.43
Trumpet	0.90	107	0.91	0.93	0.96	0.37	0.90	0.87	0.60	0.11	0.02	0.42	0.50	0.29
Alto-Sax.	1.22	131	0.96	0.99	0.98	0.40	0.76	0.71	0.14	0.09	0.50	0.08	0.48	0.48
T. Trombone	0.96	100	1.00	1.00	0.92	0.41	0.83	0.77	0.04	0.14	0.34	0.06	0.55	0.47

Additional WAE-Fader results on the playing techniques of instruments in other orchestral families

variables that can be modulated and combined. The unsupervised latent dimensions organize remaining data features, which can be directly visualized in a 3-d. space, in order to perform sampling or explicit control. These features allow to generate timbres, playing styles and hybrid effects across multiple attribute combinations through intuitive interactions. We provide a real-time implementation of our models by relying on the fine-tuned $\{G \circ \text{MCNN}\}$ generation. This implementation relies on the *LibTorch* C++ API, which converts trained *PyTorch* models, that we further embed in a *PureData* external. This plugin can be mapped to a MIDI controller or integrated in a DAW for composition and musical performance. This allows to play notes with a keyboard, while using continuous faders to control latent coordinates and mix style conditions.

Conclusion

We developed an expressive musical conditioning of the Wasserstein Auto-Encoders able to model a collection of orchestral note samples. The model learns effective target semitone and octave controls as well as continuous style variables. We considered extended playing techniques and timbre subsets as attributes, and used adversarial latent training to encourage an attribute-invariant representation in the WAE-Fader. Our ablation study validates the effectiveness of style conditioning when this invariance condition is obtained.

We fine-tuned the decoders with a Mel magnitude spectrogram inversion network that allows real-time waveform rendering and are currently working on refining the audio quality. This results in a note sample generator with meaningful data visualizations and intuitive controls of audio styles. These parameters can be mixed, as faders, in order to explore hybrid sound effects. Our final generative model is embed in a plugin for MIDI mapping and live interactions. This system provides assisted music production and fosters creative sound experimentations. We provide sound examples from our orchestral models, either inverted offline with GLA or with the fine-tuned waveform generation pipeline. These sounds allow for subjective evaluation of both semantic and audio qualities of our solution.

Although we used clearly defined metadata attributes pertaining to instrumental playing styles, the model can potentially be applied to any sound domain. For instance, a user library with custom tags could be mapped to sound synthesis parameters. Furthermore, as the architecture is rather light and scales to small datasets, it could be trained on user libraries. Future experiments will target the quality of the waveform modelling systems for variable note lengths and real-time synthesis. Ultimately, our models could be implemented as a standalone instrument with physical controls that can be mapped to pretrained style variables. This would allow an intuitive and creative exploration across a vast amount of sound variations with a reduced set of adaptive parameters.

Hierarchical timbre representations. Building on the recent advances in raw waveform modelling, the overlap-add synthesis approach with spectrogram reconstruction loss in SING [52] and the use of efficient DDSP [70] components, we propose a hierarchical VAE model which learns representations of both local acoustic features and longer temporal structure in an end-to-end architecture. We refer to this approach as Neural Granular Sound Synthesis as we draw inspiration from concatenative granular synthesizers such as [233] which we extend with probabilistic generative modelling. The bottom level VAE learns a continuous grain latent space by auto-encoding individual waveform windows of short fixed length (grains of about 90 milliseconds), whereas classical approaches to granular synthesis use hand-crafted audio descriptors to visualise the acoustic relationships within a given audio grain library. The advantage of our approach is two-fold as it learns analysis dimensions from the data observations and that these latent dimensions are continuously invertible, meaning that we can generate audio grains and interpolations from any coordinate in the VAE prior rather than performing nearest neighbour look-up in the pre-recorded grain library. Distances in the grain latent space reflect the degree of acoustic similarity between grains but do not account for the temporal dependencies in natural sounds, for instance a series of grains which result in the attack, decay and frequency modulations of instrument notes. To that end we train an upper level recurrent VAE on ordered series of grains encoded in the bottom latent space. This hierarchical model learns to embed the temporal relationships of grain features as a single code in the upper latent space which can be sampled and decoded into structured series of latent grain features. The resulting series of features is then decoded into individual waveform grains that are assembled by overlap-add.

The model training is performed in two stages by first training the bottom VAE in isolation to learn the acoustic representation of individual grains. Then we add the recurrent VAE embedding that is trained along with the bottom VAE, resulting in multiple training objectives jointly optimised during the second training phase:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{g} \mathcal{D}_{KL}\big[q_\phi(\mathbf{z}_i|\mathbf{x}_i) \parallel p_\theta(\mathbf{z})\big] + \mathcal{D}_{KL}\big[q_\phi(\mathbf{e}|\mathbf{s}_\mathbf{z}) \parallel p_\theta(\mathbf{e})\big] \\ + \frac{1}{g}\sum_{i=1}^{g}(\mathbf{z}_i - \hat{\mathbf{z}}_i)^2 + \sum_{n=1}^{N} ||l_n(\mathbf{x}) - l_n(\hat{\mathbf{x}})||_1.\end{aligned} \tag{63}$$

▷ Grain latent space regularisation: we apply the KLD between the bottom encoder

posterior $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$ and the isotropic Gaussian prior $p_\theta(\mathbf{z})$ over individual grains $\mathbf{x}_i$. For a series of g grains, the bottom encoder outputs the corresponding series of latent features $\mathbf{s_z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_g\}$.

- ▷ Temporal embedding regularisation: we apply the KLD between the recurrent encoder posterior $q_\phi(\mathbf{e}|\mathbf{s_z})$ and the isotropic Gaussian prior $p_\theta(\mathbf{e})$ over fixed-length ordered series of latent grain features. Here $\mathbf{e}$ is a single vector embedding for the whole series of grain features.

- ▷ Reconstruction in the grain latent space: we apply the MSE distance between the bottom encoder output $\mathbf{s_z}$ and the recurrent decoder output $\hat{\mathbf{s}}_\mathbf{z}$ series of latent grain features.

- ▷ Audio reconstruction: we apply the multi-scale distance [280] between magnitude spectrograms $l_n(.)$ which are computed for several STFT settings $n \in [1, N]$ by increasing hop and window sizes (i.e. decreasing the temporal resolution while increasing the frequency resolution). Here $\hat{\mathbf{x}}$ is the overlap-add output of the bottom decoder provided with latent features $\hat{\mathbf{s}}_\mathbf{z}$ and $\mathbf{x}$ is the input waveform before slicing the grains $\mathbf{x}_i$.

Each waveform window is synthesised by spectral domain noise filtering such that the bottom VAE decoder outputs coefficients of the subtractive synthesizer, a method that we adapt from [70]. While the original DDSP filter module is used to generate the stochastic residual component of the audio as short non-overlapping windows, our module aims at jointly modelling both deterministic and stochastic components using a larger window size with 75% overlap for an increased spectral resolution and smooth output audio. In addition, we train an output convolution for post-processing of the generated audio similarly to [59]. We choose this approach for the flexibility to model non-harmonic and unpitched sounds such as drum kits, to that end we run experiments on several diverse datasets including SOL strings (sr=22050Hz), one-shots in 8 drum classes (sr=16000Hz) and the 10 animal sound classes of the ESC-50 dataset (sr=22050Hz). For datasets sampled at 22050Hz, we choose a window size of 2048 and a sample duration of about 1.6 second (overlap-add of 65 grains). For datasets sampled at 16000Hz, we choose a window size of 1024 and a sample duration of one second (the trimmed one-shot length).

We evaluate several variations of the model, the baselines use a waveform decoder with either 1-dimensional transposed convolutions or nearest-neighbour interpolation followed by regular unit stride convolutions. We compare the baselines reconstruction accuracy against the decoder with subtractive noise filtering and with the added post-processing module. We observe that the DDSP-inspired decoders consistently outperform regular convolution based decoders while training about 6 times faster and sampling one second of audio in around 25 milliseconds. Based on our best performing model with subtractive noise filtering and learned post-processing output convolution, we propose several interactions for neural audio synthesis. In the first place, we can generate variable-length audio as continuous interpolations in the grain latent space of the bottom VAE. This resembles some of the usual techniques of granular sound synthesis and generates smooth acoustic textures (no specific temporal structure), with the benefit of a continuously invertible grain space from which we can freely sample at any position and step size along the user-specified latent trajectories. One such example is to use looped trajectories (e.g. circle on a n-sphere or forward-backward along a vector) which give a sense of motion to the synthesised audio and that can be repeated seamlessly. In the second place, we take advantage of the learned temporal embedding to sample grain series with conditioning on either note classes or drum classes which are decoded into fixed-length audio with spectro-temporal structures such as the attack, decay and dynamic spectrum of one-shot drum samples. Moreover it allows morphing audio samples by linear interpolation between temporal embedding vectors $\mathbf{e}_\alpha = \alpha * \mathbf{e}_2 + (1 - \alpha) * \mathbf{e}_1 \ \forall \alpha \in [0, 1]$, for instance generating kick drums with varying amounts of sustain.

As a demonstration prototype, we implement the drum class conditional model into a neural drum machine with a multi-track step sequencer in MaxMSP[13] and Python OSC[14]. Some additional visualisations, audio samples and videos are hosted on the dedicated online repository: `https://adrienchaton.github.io/neural_granular_synthesis/`. This work was accepted to the International Computer Music Conference (ICMC-2020), however due to the current pandemic the conference is re-scheduled in 2021 and should feature selected works from both years[15].

[13] `https://cycling74.com`
[14] `https://pypi.org/project/python-osc/`
[15] `http://icmc2021.org`

Neural Granular Sound Synthesis

Adrien Bitton, Philippe Esling & Tatsuya Harada
(original publishing template of ICMC 2020 available at
https://arxiv.org/abs/2008.01393)

Granular sound synthesis is a popular audio generation technique based on rearranging sequences of small waveform windows. In order to control the synthesis, all grains in a given corpus are analyzed through a set of acoustic descriptors. This provides a representation reflecting some form of local similarities across the grains. However, the quality of this grain space is bound by that of the descriptors. Its traversal is not continuously invertible to signal and does not render any structured temporality.
We demonstrate that generative neural networks can implement granular synthesis while alleviating most of its shortcomings. We efficiently replace its audio descriptor basis by a probabilistic latent space learned with a Variational Auto-Encoder. A major advantage of our proposal is that the resulting grain space is invertible, meaning that we can continuously synthesize sound when traversing its dimensions. It also implies that original grains are not stored for synthesis. To learn structured paths inside this latent space, we add a higher-level temporal embedding trained on arranged grain sequences.
The model can be applied to many types of libraries, including pitched notes or unpitched drums and environmental noises. We experiment with the common granular synthesis processes and enable new ones.

Introduction

The process of generating musical audio has seen a continuous expansion since the advent of digital systems. Audio synthesis methods relying on parametric models can be derived from physical considerations, spectral analysis (sinusoids plus noise [238] models) or signal processing operations (frequency modulation). Alternatively to those signal generation techniques, samplers provide synthesis mechanisms by relying on stored waveforms and sets of audio transformations. However, when tackling large audio sample libraries, these methods cannot scale and are also unable to aggregate a model over the whole data. Therefore, they cannot globally manipulate the audio features in the sound generation process. To this extent, corpus-based synthesis has been introduced by slicing sets of signals in shorter audio segments, which can be rearranged into new waveforms through a selection algorithm.

An instance of corpus-based synthesis, named *granular sound synthesis* [224], uses short

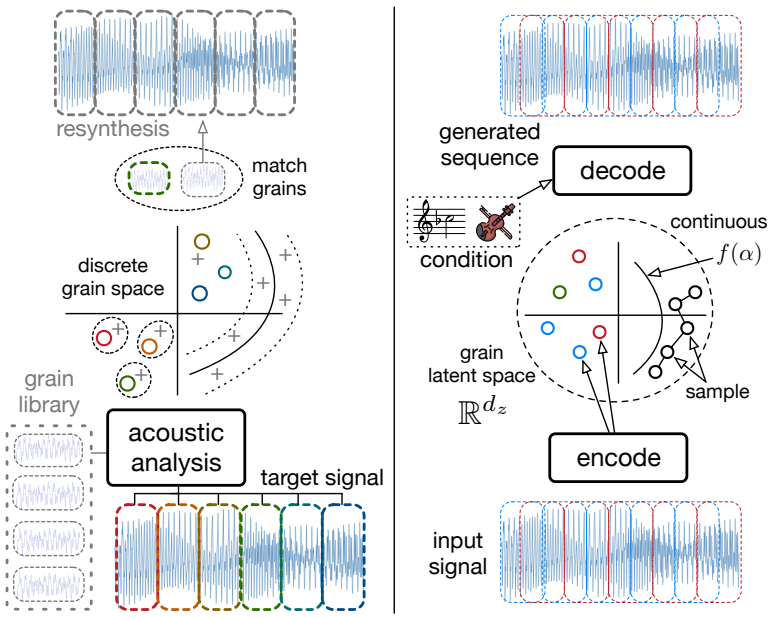

Figure 34: Left: A grain library is analysed and scattered (+) into the acoustic dimensions. A target is defined, by analysing an other signal (o) or as a free trajectory, and matched to the library through the acoustic descriptors. Subsequently, grains are selected and arranged into a waveform. Right: The grain latent space can continuously synthesize waveform grains. Latent features can be encoded from an input signal, sampled from a structured temporal embedding or freely drawn. Explicit controls can be learned as target conditions for the decoder.

waveform windows of a fixed length. These units (called *grains*) usually have a size ranging between 10 and 100 milliseconds. For a given corpus, the grains are extracted and can be analyzed through audio descriptors [203] in order to facilitate their manipulation. Such analysis space provides a representation that reflects some form of local similarities across grains. The grain corpus is displayed as a cloud of points whose distances relate to some of their acoustic relationships. By relying on this space, resynthesis can be done with *concatenative sound synthesis* [234]. To a certain extent, this process can emulate the spectro-temporal dynamics of a given signal. However, the perceptual quality of the audio similarities, assessed through predefined sets of acoustic descriptors, is inherently biased by their design. These only offer a limited consistency across many different sounds, within the corpus and with respect to other targets. Furthermore, it should be noted that the synthesis process can only use the original grains, precluding continuously invertible interpolations in this grain space.

To enhance the expressivity of granular synthesis, grain sequences should be drawn in more flexible ways, by understanding the temporal dynamics of trajectories in the acoustic descriptor space. However, current methods are only restricted to perform random or simple hand-drawn paths. Traversals across the space map to grain series that are ordered according

to the corresponding feature. However, given that the grain space from current approaches is not invertible, these paths do not correspond to continuous audio synthesis, besides that of each of the scattered original grains. This could be alleviated by having a denser grain space (leading to smoother assembled waveform), but it would require a correspondingly increasing amount of memory, quickly exceeding the gigabyte scale when considering nowadays sound sample library sizes. In a real-time setting, this causes further limitations to consider in a traditional granular synthesis space. As current methods only account for local relationships, they cannot generate the structured temporal dynamics of musical notes or drum hits without having a strong inductive bias, such as a target signal. Finally, the audio descriptors and the slicing size of grains are critical parameters to choose for these methods. They model the perceptual relationships across elements and set a trade-off: shorter grains allow for a denser space and faster sound variations at the expense of a limited estimate of the spectral features and the need to process larger series for a given signal duration.

In this paper, we show that we can address most of the aforementioned shortcomings by drawing parallels between granular sound synthesis and probabilistic latent variable models. We develop a new neural granular synthesis technique that refines granular synthesis and is efficiently solved by generative neural networks. Through the repeated observation of grains, our proposed technique adaptively and unsupervisedly learns analysis dimensions, structuring a latent grain space, which is continuously invertible to signal domain. Such space embeds the training dataset, which is no longer required in memory for generation. It allows to continuously generate novel grains at any interpolated latent position. In a second step, this space serves as basis for a higher-level temporal modeling, by training a sequential embedding over contiguous series of grain features. As a result, we can sample latent paths with a consistent temporal structure and moreover relieve some of the challenges to learn to generate raw waveforms. Its architecture is suited to optimizing local spectro-temporal features that are essential for audio quality, as well as longer-term dependencies that are efficiently extracted from grain-level sequences rather than individual waveform samples. The trainable modules used are well-grounded in digital signal processing (DSP), thus interpretable and efficient for sound synthesis. By providing simple variations of the model, it can adapt to many audio domains as well as different user interactions. With this motivation, we report several experiments applying the creative potentials of granular synthesis to neural waveform modeling: continuous free-synthesis with variable step size, one-shot sample generation with controllable attributes, analysis/resynthesis for audio style transfer and high-level interpolation between audio samples.

State of the art

Generative neural networks. *Generative models* aim to understand a given set $\mathbf{x} \in \mathbb{R}^{d_x}$ by modeling an underlying probability distribution $p(\mathbf{x})$ of the data. To do so, we consider *latent variables* defined in a lower-dimensional space $\mathbf{z} \in \mathbb{R}^{d_z}$ ($d_z \ll d_x$), as a higher-level representation *generating* any given example. The complete model is defined by $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. However, a real-world dataset follows a complex distribution that cannot be evaluated analytically. The idea of *variational inference* (VI) is to address this problem through *optimization* by assuming a simpler distribution $q_\phi(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}$ from a family of approximate densities [145]. The goal of VI is to minimize differences between the approximated and real distribution, by using their Kullback-Leibler (KL) divergence

$$q_\phi^*(\mathbf{z}|\mathbf{x}) = \underset{q_\phi(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}}{\operatorname{argmin}} \mathcal{D}_{KL}\big[q_\phi\left(\mathbf{z}|\mathbf{x}\right) \parallel p_\theta\left(\mathbf{z}|\mathbf{x}\right)\big].$$

By developing this divergence and re-arranging terms (detailed development can be found in [145]), we obtain

$$\log p(\mathbf{x}) - \mathcal{D}_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})\big] = \mathbb{E}_{\mathbf{z}}\big[\log p(\mathbf{x}|\mathbf{z})\big] - \mathcal{D}_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big].$$

This formulation of the *Variational Auto-Encoder* (VAE) relies on an encoder $q_\phi(\mathbf{z}|\mathbf{x})$, which aims at minimizing the distance to the unknown conditional latent distribution. Under this assumption, the Evidence Lower Bound Objective (ELBO) is optimized by minimization of a β weighted KL regularization over the latent distribution added to the reconstruction cost of the decoder $p_\theta(\mathbf{x}|\mathbf{z})$

$$\mathcal{L}_{\theta,\phi} = \underbrace{-\mathbb{E}_{q_\phi(\mathbf{z})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big]}_{\text{reconstruction}} + \beta * \underbrace{\mathcal{D}_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big]}_{\text{regularization}}.$$

The second term of this loss requires to define a prior distribution over the latent space, which for ease of sampling and back-propagation is chosen to be an isotropic gaussian of unit variance $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Accordingly, a forward pass of the VAE consists in *encoding* a given data point $q_\phi : \mathbf{x} \to \{\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}(\mathbf{x})\}$ to obtain a mean $\boldsymbol{\mu}(\mathbf{x})$ and variance $\boldsymbol{\sigma}(\mathbf{x})$. These allow us to obtain the latent $\mathbf{z}$ by sampling from the Gaussian, such that $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}(\mathbf{x}))$.

The representation learned with a VAE has a smooth topology [111] since its encoder is regularized on a continuous density and intrinsically supports sampling within its unsupervised training process. Its latent dimensions can serve both for analysis when encoding new samples, or as generative variables that can continuously be decoded back to the target data domain.

Furthermore, it has been shown [72] that it could be successfully applied to audio generation. Thus, it is the core of our neural model for granular synthesis of raw waveforms.

Neural waveform generation. Applications of generative neural networks to raw audio data must face the challenge of modeling time series with very high sampling rates. Hence, the models must account for both local features ensuring the generated audio quality, as well as longer-term relationships (consistent over tens of thousands of samples) in order to form meaningful signals. The first proposed approaches were based on auto-regressive models, which exploit the causal nature of audio. Given the whole waveform $\mathbf{x} = \{x_1, \ldots, x_T\}$, these models decompose the joint distribution into a product of conditional distributions. Hence, each sample is generated conditionally on all previous ones

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1}).$$

Amongst these models, WaveNet [268] has been established as the reference solution for high-quality speech synthesis. It has also been successfully applied to musical audio with the Nsynth dataset [71]. However, generating a signal in an auto-regressive manner is inherently slow since it iterates one sample at a time. Moreover, a large convolutional structure is needed in order to infer even a limited context of 100ms. This results in heavy models, only adapted to large databases and requiring long training times.

Specifically for musical audio generation, the Symbol-to-Instrument Neural Generator (SING) proposes an overlap-add convolutional architecture [52] on top of which a sequential embedding S is trained on frame steps $\mathbf{F}_{1\ldots f}$, by conditioning over instrument, pitch and velocity classes $(\mathbf{I}, \mathbf{P}, \mathbf{V})$. The model processes signal windows of 1024 points with a 75% overlap, thus reducing the temporal dimension by 256 before the forward pass of the up-sampling convolutional decoder D. Given an input signal with log-magnitude spectrogram $l(\mathbf{x}) = \log(\epsilon + |\text{STFT}[\mathbf{x}]|^2)$, the decoder outputs a reconstruction $\hat{\mathbf{x}}$, in order to optimize

$$\underset{D,S}{\text{argmin}} \; ||l(\mathbf{x}) - l(\hat{\mathbf{x}})||_1$$

for $\hat{\mathbf{x}} = D(S(\mathbf{F}, \mathbf{I}, \mathbf{P}, \mathbf{V}))$. This approach removes auto-regressive computation costs and offers meaningful controls, while achieving high-quality synthesis. However, given its specific architecture, it does not generalize to generative tasks other than sampling individual instrumental notes of fixed duration in pitched domains.

Recently, additional inductive biases arising from digital signal processing have allowed

to specify tighter constraints on model definitions, leading to high sound quality with lower training costs. In this spirit, the Neural Source-Filter (NSF) model [280] applies the idea of Spectral Modeling Synthesis (SMS) [238] to speech synthesis. Its input module receives acoustic features and computes conditioning information for the source and temporal filtering modules. In order to render both voiced and unvoiced sounds, a sinusoidal and gaussian noise excitations are fed into separate filter modules. Estimation of noisy and harmonic components is further improved by relying on a multi-scale spectrogram reconstruction criterion.

Similar to NSF, but for pitched musical audio, the Differentiable Digital Signal Processing [70] model has been proposed. Compared to NSF, this architecture features an harmonic additive synthesizer that is summed with a subtractive noise synthesizer. Envelopes for the fundamental frequency and loudness as well as latent features are extracted from a waveform and fed into a recurrent decoder which controls both synthesizers. An alternative filter design is proposed by learning frequency-domain transfer functions of time-varying Finite Impulse Response (FIR) filters. Furthermore, the summed output is fed into a reverberation module that refines the acoustic quality of the signal. Although this process offers very promising results, it is restricted in the nature of signals that can be generated.

Neural granular sound synthesis

In this paper, we propose a model that can learn both a local audio representation and modeling at multiple time scales, by introducing a neural version of the *granular sound synthesis* [234]. The audio quality of short-term signal windows is ensured by efficient DSP modules optimized with a spectro-temporal criterion suited to both periodic and stochastic components. We structure the relative acoustic relationships in a latent grain space, by explicitly reconstructing waveforms through an *overlap-add mechanism* across audio grain sequences. This synthesis operation can model any type of spectrogram, while remaining interpretable. Our proposal allows for analysis prior to data-driven resynthesis and also performs continuous variable length free-synthesis trajectories. Taking advantage of this grain-level representation, we further train a higher-level sequence embedding to generate audio events with meaningful temporal structure. In its less restrictive definition, our model allows for unconditional sampling, but it can be trained with additional independent controls (such as pitch or user classes) for more explicit interactions in composition and sound transfer.

Latent grain space. Formally, we consider a set $\mathcal{X}$ of audio grains $\mathbf{x}_i \in \mathbb{R}^{d_x}$ extracted from audio waveforms $\mathbf{x}$ in a given sound corpus, with fixed grain size d_x. This set of grains

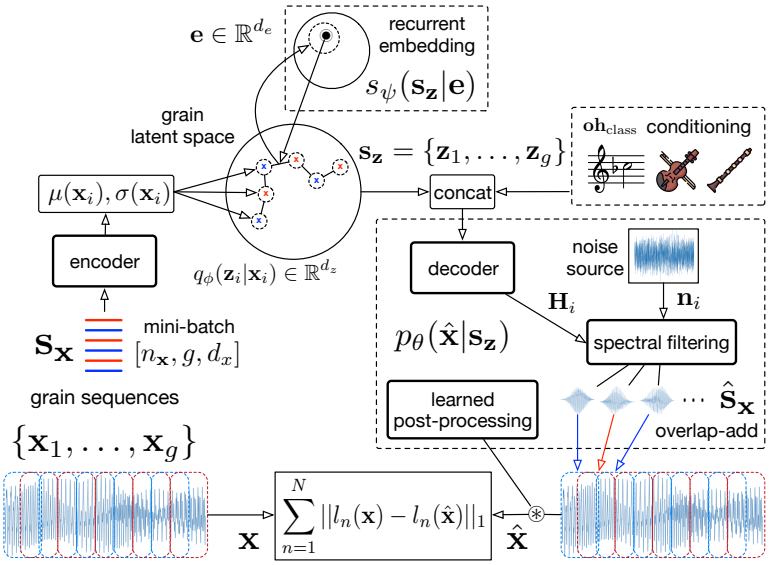

Figure 35: Overview of the neural granular sound synthesis model.

follows an underlying probability density $p(\mathbf{x}_i)$ that we aim to approximate through a parametric distribution p_θ. This would allow to synthesize consistent novel audio grains by sampling $\hat{\mathbf{x}}_j \sim p_\theta(\mathbf{x}_i)$. This likelihood is usually intractable, we can tackle this process by introducing a set of latent variables $\mathbf{z} \in \mathbb{R}^{d_z}$ ($d_z \ll d_x$). This low-dimensional space is expected to represent the most salient features of the data, which might have led to generate a given example. In our case, it will efficiently replace the use of acoustic descriptors, by optimizing continuous generative features. This latent grain space is based on an encoder network that models $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$ paired with a decoder network $p_\theta(\mathbf{x}_i|\mathbf{z}_i)$ allowing to recover $\hat{\mathbf{x}}_i$ for every grains $\mathbf{x}_i \in \mathcal{X}$. We use the Variational Auto-Encoder [145] with a mean-field family and Gaussian prior to learn a smooth latent distribution $p(\mathbf{z})$.

Latent path encoder. As we will perform overlap-add reconstruction, our model processes series of g grains $\mathbf{s_x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_g\}$ extracted from a given waveform $\mathbf{x}$. The down-sampling ratio between the waveform duration T and number of grains g is given by the hop size separating neighboring grains. Each of these grains $\mathbf{x}_i$ is analyzed separately by the encoder in order to produce $q_\phi(\mathbf{z}_i|\mathbf{x}_i) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}_i), \boldsymbol{\sigma}(\mathbf{x}_i))$. Hence, the successive encoded grains form a corresponding series $\mathbf{s_z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_g\}$ of latent coordinates such that

$$\mathbf{z}_i = \boldsymbol{\mu}(\mathbf{x}_i) + \boldsymbol{\varepsilon} * \boldsymbol{\sigma}(\mathbf{x}_i)$$

with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The layers of the encoder are first strided residual convolutions that successively down-sample the input grains through temporal 1-dimensional filters. The output

of these layers is then fed into several fully-connected linear layers that map to Gaussian means and variances at the desired latent dimensionality d_z.

Spectral filtering decoder. Given a latent series $\mathbf{s_z}$, the decoder must first synthesize each grain prior to the overlap-add operation. To that end, we introduce a filtering model that adapts the design of [70] to granular synthesis. Hence, each $\mathbf{z}_i$ is processed by a set of residual fully-connected layers that produces frequency domain coefficients $\mathbf{H}_i \in \mathbb{R}^{d_h}$ of a filtering module that transforms uniform noise excitations $\mathbf{n}_i \sim \mathcal{U}^{d_x}_{[-1,1]}$ into waveform grains. We replace the recurrence over envelope features proposed in [70] by performing separate forward passes over overlapping grain features. Denoting the Discrete Fourier Transform DFT and its inverse iDFT, this amounts to computing

$$\hat{\mathbf{X}}_i = \mathbf{H}_i * \mathrm{DFT}(\mathbf{n}_i)$$
$$\hat{\mathbf{x}}_i = \mathrm{iDFT}(\hat{\mathbf{X}}_i).$$

Since the DFT of a real valued signal is Hermitian, symmetry implies that for an even grain size d_x, the network only filters the $d_h = d_x/2 + 1$ positive frequencies.

These grains are then used in an overlap-add mechanism that produces the waveform, which is passed through a final learnable post-processing inspired from [59]. This module applies a multi-channel temporal convolution that learns a parallel set of time-invariant FIR filters and improves the audio quality of the assembled signal $\hat{\mathbf{x}}$.

Sequence trajectories embedding. As argued earlier, generative audio models need to sample audio events with a consistent long-term temporal structure. Our model provides this in an efficient manner, by learning a higher-level distribution of sequences $s_\psi(\mathbf{s_z})$ that models temporal trajectories in the granular latent space $\mathbf{s_z} \in \mathbb{R}^{d_z * g}$. This allows to use the down-sampling of an intermediate frame-level representation in order to learn longer-term relationships. This is achieved by training a temporal recurrent neural network on ordered sequences of grain features $\mathbf{s_z}$. This process can be applied equivalently to any types of audio signals. As a result, our proposal can also synthesize and transfer meaningful temporal paths inside the latent grain space. It starts by sampling $\mathbf{e} \in \mathbb{R}^{d_e}$ from the Gaussian $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then sequentially decoding $s_\psi(\mathbf{s_z}|\mathbf{e})$ and finally generating the grains and overlap-add waveform with $p_\theta(\hat{\mathbf{x}}|\mathbf{s_z})$.

Multi-scale training objective. To optimize the waveform reconstruction, we rely on a multi-scale spectrogram loss [280, 70], where STFTs are computed with increasing hop and window sizes, so that the temporal scale is down-sampled while the spectral accuracy is refined. We use both linear and log-frequency STFT [36] on which we compare log-magnitudes $l(\mathbf{x}) = \log(\epsilon + |\text{STFT}[\mathbf{x}]|^2)$ with the L1 distance $||.||_1$. In addition to fitting multiple resolutions of $\text{STFT}_{1...N}$, we can explicitly control the trade-off between low and high-energy components with the ϵ floor value [52]. In order to optimize a latent grain space, KL regularization and sampling are performed for each latent point $\mathbf{z}_i$, thus we extend the original VAE objective as

$$\mathcal{L}_{\theta,\phi} = \underbrace{\sum_{n=1}^{N} ||l_n(\mathbf{x}) - l_n(\hat{\mathbf{x}})||_1}_{\text{reconstructions}} + \beta * \underbrace{\sum_{i=1}^{g} \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}_i|\mathbf{x}_i) \parallel p_\theta(\mathbf{z})\right]}_{\text{regularizations}}$$

where N is the number of scales in the spectrogram loss and g is the number of grains processed in one sequence.

Experiments

Datasets. In order to evaluate our model across a wide variety of sound domains, we train on the following datasets

1. *Studio-On-Line* provides individual note recordings sampled at 22050 Hz with labels (pitch, instrument, playing technique) for 12 orchestral instruments. The tessitura for *Alto-Saxophone, Bassoon, Clarinet, Flute, Oboe, English-Horn, French-Horn, Trombone, Trumpet, Cello, Violin, Piano* are in average played in 10 different extended techniques. The full set amounts to around 15000 notes [12].

2. *8 Drums* around 6000 one-shot recordings sampled at 16000 Hz in *Clap, Cowbell, Crash, Hat, Kick, Ride, Snare, Tom* instrument classes[16].

3. *10 animals* contains around 3 minutes of recordings sampled at 22050 Hz for each of *Cat, Chirping Birds, Cow, Crow, Dog, Frog, Hen, Pig, Rooster, Sheep* classes of the ESC-50 dataset[17].

For datasets sampled at 22050 Hz, we use a grain size $d_x = 2048$, which subsequently sets the filter size $d_h = 1025$, and compute spectral losses for STFT window sizes $[128, 256, 512, 1024, 2048]$.

[16] `https://github.com/chrisdonahue/wavegan`
[17] `https://github.com/karolpiczak/ESC-50`

For datasets sampled at 16000 Hz, $d_x = 1024$ and STFT window sizes range from 32 to 1024. Hop sizes for both grain series and STFTs are set with an overlap ratio of 75%. Log-magnitudes are computed with a floor value $\epsilon = 5e^{-3}$. Dimensions for latent features are $d_z = 96$ and $d_e = 256$.

Models. Since datasets provide some labels, we both train unconditional models and variants with decoder conditioning. For instance *Studio-On-Line* can be trained with control over pitch and/or instrument classes when using multiple instrument subsets. Otherwise for a single instrument we can instead condition on its playing styles (such as *Pizzicato* or *Tremolo* for the *violin*). To do so, we concatenate one-hot encoded labels $\mathbf{oh}_{\text{class}}$ to the latent vectors at the input of the decoder. During generation we can explicitly set these target conditions, which provide independent controls over the considered sound attributes

$$p_\theta : (\mathbf{s_z}, \mathbf{oh}_{\text{class}}) \rightarrow \hat{\mathbf{s}}_{\mathbf{x}}^{\text{cond.}} \rightarrow \hat{\mathbf{x}}^{\text{cond.}}.$$

Training. In the first epochs only the reconstruction is optimized, which amounts to $\beta = 0$. This regularization strength is then linearly increased to its target value, during some warm-up epochs. The last epochs of training optimize the full objective at the target regularization strength, which is roughly fixed in order to balance the gradient magnitudes when individually back-propagating each term of the objective. The number of training iterations vary depending on the datasets, we use a minibatch size of 40 grain sequences, an initial learning rate of $2e^{-4}$ and the ADAM optimizer. In this setting, a model can be fitted within 10 hours on a single GPU, such as an Nvidia Titan V.

Results

The model performance is first compared to some baseline auto-encoders. To assess the generative qualities of the model, we provide audio samples of data reconstructions as well as examples of neural granular sound synthesis. These are generations based on its common processes as well as novel interactions enabled by our proposed neural architecture.

Baseline comparison. In the first place, the granular VAE could be implemented using a convolutional decoder that symmetrically reverts the latent mapping of the encoder we use. Strided down-sampling convolutions can be mirrored with transposed convolutions or up-sampling followed with convolutions. We will refer to these baselines as VAE_{tr} and VAE_{up}

while our model with spectral filtering decoder is VAE_{fi} and with the added learnable post-processing is VAE_{fi+pp}. We train these models on the *Studio-On-Line* dataset for the full orchestra in *ordinario* and the *strings* in all playing modes as well as the *8 Drums* dataset, keeping all other hyper-parameters identical. We report their test set spectrogram reconstruction scores for the Root Mean Squared Error (RMSE), Log-Spectral Distance (LSD) and their average time per training iteration. Each model was trained for about 10 hours. Accordingly, we can see that our proposal globally outperforms the convolutional decoder baselines, while training and generating fast. The latency of our model to synthesize 1 second of audio is about 19.7 ms. on GPU and 25.0 ms. on CPU.

	$\textbf{VAE}_{\textbf{tr}}$	$\textbf{VAE}_{\textbf{up}}$	$\textbf{VAE}_{\textbf{fi}}$	$\textbf{VAE}_{\textbf{fi+pp}}$
Studio-On-Line ordinario				
RMSE	6.86	6.65	6.22	**4.86**
LSD	1.60	1.62	1.29	**1.17**
Studio-On-Line strings				
RMSE	5.68	5.78	5.29	**4.07**
LSD	1.39	1.43	1.19	**1.05**
8 Drums				
RMSE	3.85	4.39	**2.65**	2.79
LSD	0.94	0.66	**0.52**	**0.52**
sec./iter	2.32	2.87	**0.54**	0.58

Report of the baseline model comparison. Bold denotes the best model for each evaluation.

Common granular synthesis processes. The audio-quality of the models trained in different sound domains can be judged by data reconstructions. It gives a sense of the model performance at auto-encoding various types of sounds. This extends to generating new sounds by sampling latent sequences rather than encoding features from input sounds. For structured one-shot samples, such as musical notes and drum hits, latent sequences are generated from the higher-level sequence embedding. For use in composition (e.g. MIDI score), this sampling can be done with conditioning over user classes such as pitch and target instrument. Since the VAE learns a continuously invertible grain space, it can as well be explored with smooth interpolations that render free-synthesis trajectories. Some multidimensional latent curves that are mapped to overlap-add grain sequences, including linear interpolations between random samples from the latent Gaussian prior, circular paths and spirals. When repeating forward and backward traversals of a linear interpolation or looping a circular curve, we can modu-

late non-uniformly the steps between latent points in order to bring additional expressivity to the synthesis. Free-synthesis can be performed at variable lengths (in multiples of g) by concatenating several contiguous latent paths.

Audio style and temporal manipulations. To perform data-driven resynthesis, a target sample is analyzed by the encoder. Its corresponding latent features are then decoded, thus emulating the target sound in the *style* of the learned grain space. A conditioning over multiple *timbres* (e.g. instrument classes) allows for finer control over such audio transfer between multiple target *styles*. To perform resynthesis of audio samples longer than the grain series length g, we auto-encode several contiguous segments that are assembled with fade-out/fade-in overlaps. Since the model can also learn a continuous temporal embedding, by interpolating this higher-level space, we can generate successive latent series in the grain space that are decoded into signals with evolving temporal structures.

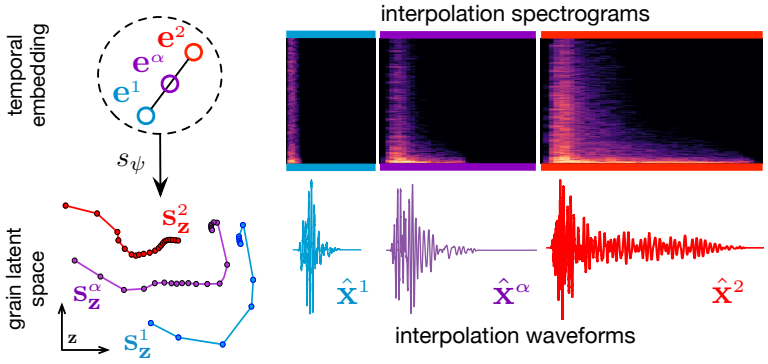

Figure 36: An interpolation in the continuous temporal embedding generates series of latent grain features corresponding to waveforms with evolving temporal structure. Here three drum sounds with increasingly sustained envelope. The point $\mathbf{e}^\alpha$ *is set half-way from* $\mathbf{e}^1$ *and* $\mathbf{e}^2$*.*

Real-time sound synthesis. With GPU support, for instance a sufficient dedicated laptop chip or an external thunderbolt hardware, the models can be ran in real-time. In order to apply trained models to these different generative tasks, we currently work on some prototype interfaces based on a *Python OSC*[18] server controlled from a *MaxMsp*[19] patch. For instance a neural drum machine featuring a step-sequencer driving a model with sequential embedding and conditioning trained over the *8 Drums* dataset classes.

[18] `https://pypi.org/project/python-osc/`
[19] `https://cycling74.com`

Conclusions

We propose a novel method for raw waveform generation that implements concepts from granular sound synthesis and digital signal processing into a Variational Auto-Encoder. It adapts to a variety of sound domains and supports neural audio modeling at multiple temporal scales. The architecture components are interpretable with respect to its spectral reconstruction power. Such VAE addresses some limitations of traditional techniques by learning a continuously invertible grain latent space. Moreover, it enables multiple modes of generation derived from granular sound synthesis, as well as potential controls for composition purpose. By doing so, we hope to enrich the creative use of neural networks in the field of musical sound synthesis.

Discrete timbre representations. Another common approach in granular sound processing [232] relies on re-synthesising a source audio with grains belonging to another library. In this process the input audio is sliced and analysed with acoustic descriptors which provide a series of target features. Given the corresponding analysis features computed over the grain library, the re-synthesis algorithm matches the input slices with grains that are assembled into an output waveform that follows the acoustic descriptor target. This concept may be seen as mosaicing [63] a library of audio grains to emulate the source audio, given a dynamic criterion computed by acoustic descriptors. To some extent, this process relates to the implicit acoustic models of timbre presented in section 4.3 which extract a compressed acoustic representation, the fundamental frequency and loudness envelopes, that is decoded into an audio with the spectral distribution of the learned timbre. These models are applied to timbre transfer by extracting acoustic envelopes from another source audio that are provided to the model as targets for the re-synthesis. However, these processes differ in the choice of the acoustic representation which the synthesis relies on. On the one hand, the implicit acoustic models use a timbre invariant representation (i.e. fundamental frequency and loudness) that can be extracted in other pitched sources and fed to the synthesis model of the learned timbre. On the other hand, the granular re-synthesis algorithm queries audio segments in a fixed library and uses hand-crafted acoustic descriptors as matching targets, which may entangle various perceptual properties including timbre similarities and the alignment of pitch and loudness.

Based on these observations, we propose to learn an acoustic representation of a single timbre in an end-to-end waveform model by vector-quantization of short-term latent features. The VQ-VAE [270] framework was applied to voice conversion, changing the perceived speaker identity, as well as unsupervised speech representation learning [44] so that the learned latent features would strongly correlate with the underlying phonemes of the trained language. In our work, we apply vector-quantization as a mean of learning a set of short-term features that decompose the acoustic distribution of the training timbre domain. We apply the model to timbre conversion (Figure 37) using the encoder to analyse another source audio that is matched with the discrete latent timbre features that are re-synthesised by the decoder. Quantisation acts here as a learned acoustic bottleneck for audio conversion based on spectral similarity, whereas the implicit acoustic models rely on pre-extracted features that are invariant to timbre. We relate this work to the granular re-synthesis and mosaicing approaches by learning

a discrete set of timbre features, which we interpret as short-term spectral patterns that can be matched and recombined into natural waveforms of the target timbre.

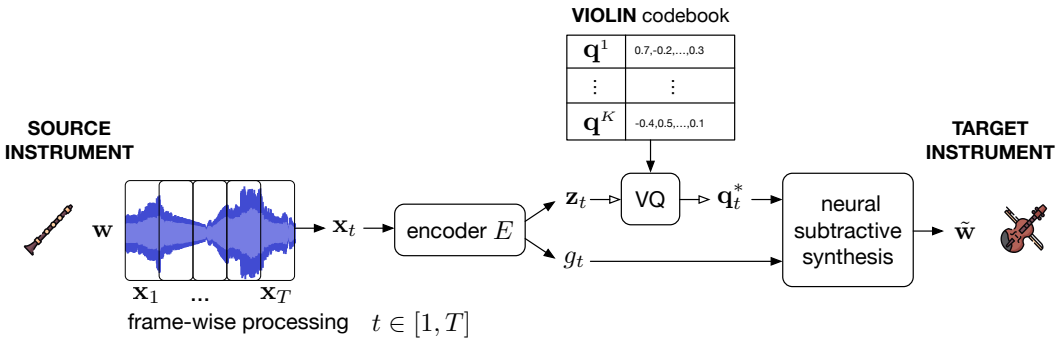

Figure 37: The proposed VQ-VAE model for variable-length timbre transfer. The source audio is encoded into series of continuous features $\mathbf{z}_t$ and spectral gains g_t, the learned latent vectors $\{\mathbf{q}^1, \ldots, \mathbf{q}^K\}$ are used for nearest-neighbour quantisation into series of matched features $\mathbf{q}_t^*$ that are passed to the subtractive synthesis decoder module, along with the predicted spectral gains. Conversion is enforced by the quantisation bottleneck, as the audio is synthesised using discrete latent features learned from the target timbre.

The model is composed of a waveform encoder with 1-dimensional convolutions that segment and down-sample series of signal windows of length 2048 with a 75% overlap ratio. These series of features are passed to two output layers, one for the continuous latent codes and a second for the spectral gains used in the decoder. The continuous latent codes are quantised by nearest neighbour lookup in the codebook embeddings (1024 vectors of 128 dimensions each) which are passed to the decoder that predicts spectral domain filtering coefficients applied to a noise source. In order to learn timbral features that are invariant to loudness, we use the additional gains predicted by the encoder to scale to spectral distributions of the filters predicted by the decoder. Accordingly, the latent codebook does not need to account for variations in audio level, although the fundamental frequency information remains embedded in the latent representation.

We train one model per instrument on variable-length monophonic performances using the separated tracks of the URMP dataset [160] down-sampled to 22050Hz. Besides the vector-quantization losses, we use the multi-scale spectrogram reconstruction and the embedding loss of [175] which we re-implemented for compatibility[20]. We compare the model against a baseline auto-encoder without vector-quantization by

[20] `https://github.com/adrienchaton/PerceptualAudio_Pytorch`

reconstruction score, transfer classification score and accuracy of the synthesised fundamental frequency and loudness envelopes. As we train the models on variable-length music performances, we pretrain a frame-wise instrument classifier which outputs a class prediction every non-overlapping frame of 4096 samples (about 185 milliseconds). Each instrument model is used to convert all the tracks of all other instruments in the database and the average classification accuracy with respect to the instrument target is reported. In addition we extract the fundamental frequency and loudness envelopes of the input audio and synthesised audio that are compared with the dynamic time warping distance to evaluate how well these features are preserved independently from timbre conversion. We observe that vector-quantization allows a significant improvement in the timbre transfer task and that the loudness accuracy is mostly improved thanks to the separate spectral gain prediction. This comes at the expense of a slight decrease in reconstruction accuracy and fundamental frequency alignment, which is caused by the discrete embedding of spectral features that comprise the fundamental frequency information.

We apply the VQ-VAE timbre model to more diverse datasets in order to show the flexibility and robustness of the approach. In the first place, we train models on singing voice using the recordings of VocalSet [287] and perform conversion from and to orchestral instrument timbres. As we do not rely on pre-extracted features such as a fundamental frequency estimate, as in DDSP that uses the pretrained CREPE prediction, we are also able to convert from much dissimilar domains which do not have a musical pitch. As an example, we use vocal imitations from the VocalSketch database [32] which are converted into performances of orchestral instruments. By this mean, we experiment with the possibility of voice-driven musical sound synthesis as an intuitive way of traducing musical ideas into sound. These vocal sketches were crowdsourced by asking untrained participants to express diverse sound concepts (e.g. object noises, moods) simply using their voice, without providing any reference audio. This process is highly relevant to music creativity as many sound ideas are hardly described in terms of usual language and notations whereas voice is a natural medium that does not require a particular music training (e.g. tuning by hand a synthesizer). Thus voice-driven synthesis appears as an intuitive interaction for musical synthesis and untrained users that can conveniently mimic a target sound synthesis idea by vocal imitation. Some additional visualisations and audio samples are hosted on the dedicated online repository: `https://adrienchaton.github.io/VQ-VAE-timbre/`.

Vector-Quantized Timbre Representation

Adrien Bitton, Philippe Esling & Tatsuya Harada

(original publishing template available at `https://arxiv.org/abs/2007.06349`)

Timbre is a set of perceptual attributes that identifies different types of sound sources. Although its definition is usually elusive, it can be seen from a signal processing viewpoint as all the spectral features that are perceived independently from pitch and loudness. Some works have studied high-level timbre synthesis by analyzing the feature relationships of different instruments, but acoustic properties remain entangled and generation bound to individual sounds. This paper targets a more flexible synthesis of an individual timbre by learning an approximate decomposition of its spectral properties with a set of generative features. We introduce an auto-encoder with a discrete latent space that is disentangled from loudness in order to learn a quantized representation of a given timbre distribution. Timbre transfer can be performed by encoding any variable-length input signals into the quantized latent features that are decoded according to the learned timbre. We detail results for translating audio between orchestral instruments and singing voice, as well as transfers from vocal imitations to instruments as an intuitive modality to drive sound synthesis. Furthermore, we can map the discrete latent space to acoustic descriptors and directly perform descriptor-based synthesis.

Introduction

Timbre is a central element in musical expression and sound perception [180], which can be seen as a set of spectral properties that allows us to distinguish instruments played at the same pitch and velocity. Synthesis of musical timbre has been studied by analyzing the feature relationships between instruments. A disentangled representation of pitch and timbre was proposed in [171] which allows to generate musical notes with instrument control. Perceptual timbre relationships were explicitly modeled in [72], and latent timbre synthesis could be iteratively mapped to target acoustic variations. However, both techniques are not evaluated in the signal domain and acoustic properties remain entangled. A timbre-invariant representation of variable-length waveforms is learned in [190] to perform unsupervised translation of an instrument performance into another, which we refer to as *timbre transfer*. However, such representation is not interpretable and does not offer any controls besides selecting a target instrument class.

This paper introduces a generative model training on an individual timbre domain that allows variable-length timbre transfer of diverse audio sources and sound synthesis with direct

acoustic descriptor control. This auto-encoder with a discrete latent space that is disentangled from loudness learns the feature quantization of a given timbre distribution. Latent features are decoded into short-term spectral coefficients of a filter applied to overlapping frames of a noise excitation. This subtractive synthesis technique does not constrain the types and lengths of signals that can be processed. We perform timbre transfer by encoding any input signals into this discrete representation. The matched series of latent features is inverted into a signal which corresponds to the trained timbre domain. Since the model has learned an approximate decomposition of a timbre into a set of short-term spectral features, we can individually decode each latent vector and compute the corresponding acoustic properties. It provides a direct mapping for *descriptor-based* synthesis. A descriptor target can be matched with a series of latent features and decoded into a signal with the desired auditory property.

Our timbre transfer experiments apply to orchestral instruments and singing voice. We pretrain an instrument classifier and evaluate transfer with the predicted accuracy of a model at translating all other instruments into the trained timbre domain. And we measure the distances between the input and output fundamental frequency and loudness. These distances amount to the error of a model at preserving the source pitch and loudness independently from transforming the timbre. We also perform timbre transfer from vocal imitations to instruments as an example of voice driven synthesis. Whereas many sound ideas are hardly described with musical parameters, which require an expert knowledge, human voice control can be an intuitive medium [300]. For instance, mimicking some moods, objects or actions that are translated into musical sounds.

State of the art

Generative Modeling. Generative neural networks aim to model a given set of observations $\mathbf{x} \in \mathbb{R}^{d_x}$ in order to consistently produce novel samples $\tilde{\mathbf{x}}$. To this extent, we introduce latent variables $\mathbf{z} \in \mathbb{R}^{d_z}$ defined in a lower-dimensional space ($d_z < d_x$). These latent features form a simpler representation from which the data can be generated. An unsupervised approach to learn these variables is the *auto-encoder*. A deterministic encoder maps observations to latent codes $\mathbf{z} = E_\phi(\mathbf{x})$ that are fed to the decoder which in turn reconstructs the input $\tilde{\mathbf{x}} = D_\theta(\mathbf{z})$. Their parameters jointly optimize some reconstruction loss

$$\underset{\phi,\theta}{\operatorname{argmin}} \, \mathcal{L}_{rec.}\left(\mathbf{x}, D_\theta\left(E_\phi\left(\mathbf{x}\right)\right)\right).$$

As this approach explicitly performs dimensionality reduction, these latent variables can extract the most salient features in the dataset. Hence, they also facilitate the generation over

high-dimensional distributions. However, in this deterministic auto-encoder setting there is no guarantee that latent inference on unseen data produces meaningful codes for the decoder. In other words, these latent projections are usually scattered apart from those of the training observations, and the decoder may fail at reconstructing anything consistent besides its training domain.

Regularized auto-encoders tackle this problem by introducing constraints over the distribution of latent codes and generation mechanism. To do so, the Variational Auto-Encoder (VAE)[145] sets a probabilistic framework by optimizing a variational approximation of the encoder distribution $q_\phi(\mathbf{z}|\mathbf{x})$ given a continuous prior $p_\theta(\mathbf{z})$ over latent variables. The model is trained with a Kullback-Leibler (KL) divergence regularizer added to a reconstruction cost

$$\mathcal{L}_{\theta,\phi} = \underbrace{-\mathbb{E}_{q_\phi(\mathbf{z})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big]}_{\text{reconstruction}} + \underbrace{\mathcal{D}_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big]}_{\text{regularization}}.$$

VAEs provide several desirable features such as their interpolation quality, generalization power from small datasets, and the ease for both high-level visualization and sampling. However, they tend to produce less detailed low-level features (blurriness effect), and the regularization can degenerate into an uninformative latent representation (a phenomenon known as *posterior collapse* [168]).

The Vector-Quantized VAE (VQ-VAE)[270] addresses these issues by learning a *discrete* latent representation, defined as a *codebook* with a fixed number of latent vectors $\{\mathbf{q}^1, \dots, \mathbf{q}^K\}$. Hence, the output $\mathbf{z}$ of the deterministic encoder is matched to its nearest embedding code $\mathbf{q}^*$

$$\mathbf{q}^* = \underset{j \in [1,K]}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{q}^j\|_2$$

which is passed to the decoder, so that it optimizes generation solely using the current codebook state. In addition to the latent dimensionality reduction, the amount of information compression is set by the size K of the discrete embedding. Assuming a uniform prior distribution over the embedding, the amount of information encoded in the representation corresponds to a constant KL divergence of $\log(K)$. Since that hyperparameter is not optimized, the VQ-VAE alleviates posterior collapse. The representation is optimized with a *codebook update* loss which matches the selected code to the encoder features

$$\mathcal{L}_{codebook} = \|sg(\mathbf{z}) - \mathbf{q}^*\|_2^2$$

where sg denotes a *stop gradient* operation, bypassing the variable in the back-propagation.

Symmetrically, the *encoder commitment* to the selected code is applied as a loss

$$\mathcal{L}_{commit} = \|\mathbf{z} - sg(\mathbf{q}^*)\|_2^2$$

in order to bound its outputs and stabilize the training. The complete objective with commitment cost β is then

$$\mathcal{L}_{\text{VQ-VAE}} = \mathcal{L}_{rec.}(\mathbf{x}, \tilde{\mathbf{x}}) + \mathcal{L}_{codebook} + \beta * \mathcal{L}_{commit}.$$

Because of the argmin operator, the nearest-neighbor quantization is not differentiable and the encoder cannot be directly optimized. However, this issue is circumvented by simply copying the gradient from $\mathbf{q}^*$ to $\mathbf{z}$ (straight-through approximation) and back-propagating this information in the encoder unaltered with respect to the quantization output. The VQ-VAE achieves sharper reconstructions than those of the probabilistic VAE, and its discrete latent representation was successfully applied to speech for unsupervised acoustic unit discovery[44]. In this paper, it was shown that the quantized codebook could extract high-level interpretable audio features that strongly correlate to phonemes, with applications for voice conversion. Inference is performed by quantizing every continuous encoder outputs with the learned latent codebook. Consequently, the decoder is bound to reconstruct the input given this discrete latent space, whose degrees of freedom can be adjusted with the codebook size K. This reconstruction with latent quantization may be seen as a transfer when matching any out-of-domain inputs with a set of features learned from a given dataset.

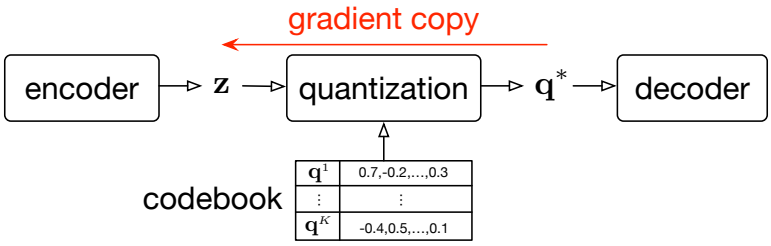

Figure 38: Overview of a Vector-Quantization (VQ) layer.

Raw Waveform Modeling. The first methods for neural waveform synthesis have relied on auto-regressive sample predictions, as in the reference WaveNet model [268]. It achieves high-fidelity sound synthesis, at the cost of a heavy architecture that is inherently slow to train and sample from. In more recent developments, waveform models have exploited digital signal processing knowledge, providing efficient solutions that achieve competitive audio quality. It results in more interpretable and lighter architectures which consequently require less data to train on. A sinusoids plus stochastic decomposition[237] is first used in the Neural

Source Filter (NSF) [280] model. It generates speech from acoustic features and the estimated fundamental frequency f_0, that are used as a conditioning information for the synthesis modules. These are a sinusoidal source controlled by the f_0, a Gaussian noise source and two separate temporal filters to process each of them. The generated signals are adaptively mixed in order to render both voiced (periodic) and unvoiced (aperiodic) speech components. More specific to musical sound synthesis, the Differentiable Digital Signal Processing (DDSP)[70] model implements a similar decomposition with an additive synthesizer conditioned with f_0 summed with a subtractive noise synthesizer, both controlled by the decoder. It predicts the harmonic amplitudes and the frequency domain coefficients $\mathbf{H}_t$ to generate the filtered audio $\mathbf{y}_t$ from non-overlapping frames of noise $\mathbf{x}_t$

$$\mathbf{y}_t = \mathrm{DFT}^{-1}(\mathbf{H}_t.\mathrm{DFT}(\mathbf{x}_t))$$

with DFT the Discrete Fourier Transform and DFT^{-1} its inverse. This model offers promising results and an interesting modularity that disentangles harmonic, stochastic as well as reverberation features. However, it is mainly tailored for harmonic sounds and does not allow end-to-end training as it relies on an external f_0 estimator.

The two aforementioned models train on a multi-scale Short-Term Fourier Transform (STFT) reconstruction objective, that is computed for several resolutions. The distance between spectrogram magnitudes is an efficient criterion for optimizing waveform reconstruction as it provides a structured time-frequency representation. However, since the phase is discarded, it may fail at evaluating certain acoustic errors. Based on human ratings to evaluate just-noticeable distortions, a differentiable audio metric is proposed [175] in order to assess artifacts at the threshold of perception. Listeners were asked whether pairs of audio were exactly similar, with one element being applied varying strengths of additive noises, reverberation or equalization. This dataset provides pairs of waveforms along with binary ratings, on which a convolutional neural network learns a differentiable loss. A deep feature distance d is trained by forwarding each audio $\mathbf{x}$ (clean) and $\tilde{\mathbf{x}}$ (altered) into the network. Considering L layers and $F_l \in \mathbb{R}^{T_l * C_l}$ the l-th convolution activations, it computes

$$d(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{l=1}^{L} \frac{1}{T_l} \left\| \mathbf{w}_l \odot (F_l(\mathbf{x}) - F_l(\tilde{\mathbf{x}})) \right\|_1$$

with $\mathbf{w}_l$ a learnable weight for each of the C_l channels of width T_l. Given this deep feature distance, a low-capacity classifier is trained to infer human ratings of noticeable dissimilarity. In this setting, the network must efficiently model such just-noticeable differences in order to allow an accurate prediction. Once trained, this distance can be used as a differentiable audio

loss. It was shown to improve the performance of speech enhancement systems and may be added as an additional reconstruction objective.

Musical Timbre Transfer. The task of musical timbre transfer is to convert the identity of one sound into another, e.g. two instruments, while preserving independent features such as pitch and loudness. The model in [171] learns a representation that disentangles these features inside instrument sounds. It offers interesting visualizations and generative controls. However, it is restricted to processing individual notes of limited duration from spectrogram magnitudes. As a result, synthesis occurs with an inversion latency and is not evaluated in the signal domain.

In this work we focus instead on unsupervised transfer for variable-length waveforms, such as recorded music performances. The Universal Music Translation Network [190] proposes an architecture for multi-domain transfers, using a shared encoder paired with domain-specific decoders. The generalization of the learned representation to many domains is achieved with a latent confusion objective. It uses an adversarial classifier to enforce the domain-invariance of latent codes. The task is solved in the waveform domain by relying on multiple WaveNet models. For that reason, both training and synthesis are slow and computationally very intensive. Although it allows high-quality auto-encoding with domain selection, its latent representation does not offer more generative controls. On the other hand, more expressive and light-weight synthesis models can perform timbre manipulations with additional constraints. The DDSP model was applied to single domain transfer with independent control over pitch and loudness, but with limitations of its amortized inference.

Vector-Quantized Model for Timbre

In this paper, we introduce a waveform auto-encoder for learning a discrete representation of an individual timbre that can be used for sound transfer and descriptor-based synthesis. We merge the VQ-VAE approach with a decoder that performs subtractive noise filtering with a disentangled gain prediction. As the model is unsupervised, it can train on diverse music performance recordings and can as well process non-musical audio such as vocal imitations. The resulting latent representation decomposes spectral timbre properties, while being invariant to loudness. The model performs timbre transfer by encoding any audio sources into the loudness-invariant feature quantization which is inverted to the learned timbre. The discrete latent space can be mapped to acoustic descriptors. It allows us to order series of latent features according to a descriptor target and offers meaningful synthesis controls.

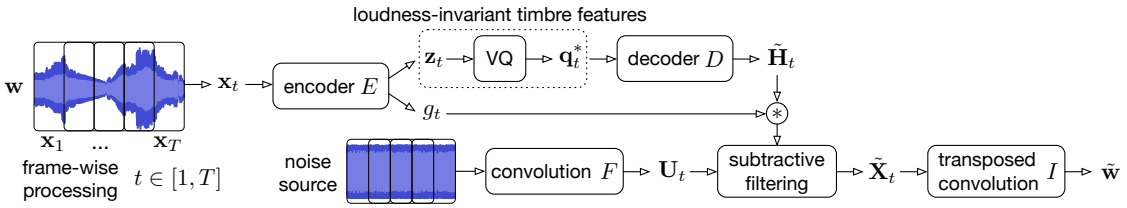

Figure 39: Architecture of our proposed Vector-Quantized subtractive sound synthesis model.

Model Overview. We define an individual timbre through a corpus of audio files recorded for a target sound domain, for instance isolated or solo performances of a given instrument. A dataset of successive overlapping signal windows $\mathbf{x}_t \in \mathbb{R}^L$ is constructed by slicing input waveforms $\mathbf{w}$ of given duration into series $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$. The encoder E projects each of the T windows of length L into a continuous latent code $E(\mathbf{x}_t) = \mathbf{z}_t \in \mathbb{R}^{d_z}$, while reducing the dimensionality as $d_z < L$. A quantization estimator selects a vector $\mathbf{q}_t^*$ in the discrete embedding $\{\mathbf{q}^1, \ldots, \mathbf{q}^K\} \in \mathbb{R}^{d_z * K}$ that is the closest match to $\mathbf{z}_t$. The decoder D predicts filtering coefficients $D(\mathbf{q}_t^*) = \tilde{\mathbf{H}}_t \in \mathbb{R}^N$ that are applied to spectral frames $\mathbf{U}_t$ of a noise excitation, with N the number of frequency bins. In order to disentangle loudness from the latent timbre embedding, the encoder predicts an additional scalar gain g_t. The output time frames are filtered as

$$\tilde{\mathbf{X}}_t = g_t * \tilde{\mathbf{H}}_t . \mathbf{U}_t .$$

The reconstruction is done by inversion of $\{\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_T\}$ into $\tilde{\mathbf{w}}$. This overlap-add uses the same stride as the encoder and the noise spectrum, and it can be performed for variable-length signals.

Encoding Modules. The first layer of the encoder slices the input waveform into overlapping windows with a convolution of stride S and Hanning kernel of size L set as a power of 2. Every individual window is passed into a stack of downsampling convolutions with stride 2. One output layer predicts the latent features $\mathbf{z}_t$ and another infers the scalar gains g_t. The latent features are projected into the discrete embedding, yielding the quantization codes $\mathbf{q}_t^*$ sent to the decoder.

Decoding Modules. Subtractive synthesis is performed by filtering an excitation with flat energy distribution. A uniform noise signal of the same length as $\mathbf{w}$ is converted into complex spectrum frames $\mathbf{U}_t$. We use a convolution F with a stride S and N kernels of size L corresponding to the Fourier basis. The first half of the bins is the real part and the other is the imaginary part. The series of quantized features $\mathbf{q}_t^*$ is processed by the decoder which

predicts the series of filtering coefficients $\tilde{\mathbf{H}}_t$. The decoder is composed of an input stack of linear layers, a Recurrent Neural Network (RNN) and an output stack of linear layers. The predicted filters are scaled with the disentangled gains as $g_t * \tilde{\mathbf{H}}_t$ and applied to the noise spectrum. Synthesis from the filtered frames $\tilde{\mathbf{X}}_t$ is done by overlap-add. We use a transposed convolution I of stride S and N kernels of size L corresponding to the inverse Fourier basis. Such use of convolutional neural networks for time-frequency analysis and synthesis has previously been detailed for both music information retrieval[36] and source separation[197] tasks.

Model Objectives. Our proposed model jointly optimizes waveform reconstruction and vector quantization with encoder commitment and codebook update losses. In order to evaluate the reconstruction, we use a multi-scale spectrogram loss over several STFT resolutions of magnitudes l_n and the deep feature distance d. The different loss contributions are scaled by hyperparameters $\lambda_{0,1}$ for reconstruction terms and λ_2 for latent optimization, as

$$\mathcal{L} = \lambda_0 * \sum_n \|l_n(\mathbf{w}) - l_n(\tilde{\mathbf{w}})\|_1 + \lambda_1 * d(\mathbf{w}, \tilde{\mathbf{w}}) + \lambda_2 * (\mathcal{L}_{codebook} + \beta * \mathcal{L}_{commit}).$$

scores	classification accuracy		DTW f_0		DTW loudness		LSD	
targets | models	baseline	VQ-VAE	baseline	VQ-VAE	baseline	VQ-VAE	baseline	VQ-VAE
bassoon	0.4256	**0.6456**	4.8e-4	**4.7e-4**	**2.5e-4**	**2.5e-4**	**0.4193**	0.4387
cello	0.3182	**0.5896**	**4.7e-4**	5.5e-4	2.2e-4	**1.9e-4**	**0.4500**	0.4853
clarinet	0.3962	**0.6811**	**4.1e-4**	4.6e-4	3.8e-4	**2.4e-4**	0.4341	**0.4303**
double-bass	0.1190	**0.4298**	6.6e-4	**6.0e-4**	**2.5e-4**	3.0e-4	**0.4313**	0.4346
flute	0.4999	**0.6765**	**6.5e-4**	8.8e-4	**2.5e-4**	2.7e-4	**0.3623**	0.3735
horn	0.4104	**0.5861**	**4.4e-4**	5.2e-4	2.4e-4	**1.8e-4**	**0.3910**	0.4409
oboe	0.6610	**0.7441**	**6.3e-4**	6.5e-4	**3.1e-4**	3.3e-4	**0.3679**	0.3840
trumpet	0.5126	**0.6041**	**3.7e-4**	4.0e-4	5.8e-4	**3.7e-4**	**0.3625**	0.3719
viola	**0.6409**	0.5689	**4.1e-4**	4.2e-4	2.4e-4	**2.3e-4**	0.4606	**0.3946**
violin	0.7434	**0.7960**	**3.7e-4**	4.5e-4	4.1e-4	**2.8e-4**	0.5546	**0.5500**
instrument average	*0.4727*	***0.6321***	***4.8e-4***	*5.4e-4*	*3.1e-4*	***2.6e-4***	***0.4233***	*0.4303*
singing	N.A.	N.A.	**3.2e-4**	3.6e-4	**2.7e-4**	2.8e-4	**0.5477**	0.5523

Score comparison of the VQ-VAE model against the baseline auto-encoder. Classification accuracy assesses the transfer to the instrument target of each model. DTW measures the distance between the source and audio transfer curves of f_0 and loudness. LSD evaluates the test set reconstruction error in the target domain. Bold denotes the best score.

Experiments

Datasets. In order to learn the individual timbre of instruments, we rely on multitrack recordings of music performances from two datasets, namely URMP[160] and Phenicx[201]. They both provide isolated audio for *bassoon*, *cello*, *clarinet*, *double-bass*, *flute*, *horn*, *oboe*, *trumpet*, *viola* and *violin*.

To learn the singing voice timbre representation, we use the recordings from the VocalSet database[287] which provides 9 female and 11 male singers individually performing several techniques and pitches. We discard the noisiest techniques *breathy*, *inhaled*, *lip-trill*, *trillo*, *vocal fry* and merge all others in the same timbre domain.

To experiment with voice-controlled sound synthesis, we use some examples of the VocalSketch database[32], which were given as source inputs to models pretrained on instruments. Vocal imitations were not used as training data, but as crowd-sourced examples of untrained human voices expressing some diverse sound concepts.

Perceptual Audio Loss. Using the dataset of just-noticeable audio differences and human ratings [175], we re-implemented the deep feature distance in *PyTorch* (codes and pretrained parameters of d are provided[21]). To use this loss as a reconstruction objective for music performances recorded at various volumes, we apply a random gain to the training audio pairs so that the learned distance is invariant to audio levels. As this criterion was trained for several perturbations including additive noises and reverberation, the model optimizes additional acoustic cues to generate audio signals that are consistent with the training dataset. We observe that vocal imitations recorded in uncontrolled conditions can be transferred into musical sounds which do not exhibit the input noise found in VocalSketch.

Training Details. All audio examples are first downsampled to 22kHz in mono format. The subsets corresponding to each individual timbre, either instrumental or singing voice, are split into training and test data (15%). We remove silences and concatenate the trimmed audio. Segments $\mathbf{w}$ of 1.5 seconds are randomly sampled in the training data and collated into mini-batches of size 20 for training the VQ-VAE. We optimize the model for 150,000 iterations with the ADAM optimizer and a learning rate of 2e-4.

The model is defined with window size $L = 2048$, stride $S = 512$ and $N = L + 2$ which corresponds to the real and imaginary parts of the halved complex spectrum. The encoder has

[21] `https://github.com/adrienchaton/PerceptualAudio_Pytorch`

7 downsampling convolutions of stride 2, with increasing output channel dimension from 32 to 256 and kernel size 13. One output layer maps to latent features of size $d_z = 128$ and another pair of linear layers outputs the scalar gains. The vector quantization space is a codebook of size $K = 1024$. The decoder has two blocks of 4 linear layers with a constant hidden dimension of 768 that are interleaved with intermediate Gated Recurrent Units of the same feature size. The output of the decoder is a linear layer that produces N filtering coefficients which are passed into a *sigmoid* activation and *log1p* compression. The convolutions F and I are initialized as the linear STFT and its inverse, future experiments could include using different frequency scales or training their kernels. The multi-scale sprectrogram reconstruction is computed for STFTs with a hop ratio of 0.25 and window sizes of [128, 256, 512, 1024, 2048]. We adjust the λ strengths in order to balance the initial gradient magnitudes of each objective, accordingly $\lambda_0 = \lambda_2 = 1$ and $\lambda_1 = 0.2$. The latent loss uses an encoder commitment strength of $\beta = 0.25$.

Classifier Model. In order to evaluate the timbre transfer task, we train a reference classifier on the 10 target instruments. We adapt the baseline proposed in [287] to perform short-term predictions rather than predicting a single label per file. Our classifier predicts a label every non-overlapping frame of 4096 samples which amounts to a context of about 185ms. The model was trained with pitch-shifting data augmentation and achieves 85% test set frame-level accuracy at predicting the correct instrument label.

Evaluation. The performance of our VQ-VAE is quantitatively compared against a *baseline* deterministic auto-encoder without vector quantization. Since its latent space is continuous, the disentangled gain prediction did not improve the baseline and is as well removed. Besides that, it shares the same encoder and decoder architectures and only optimizes reconstruction costs. We compare the models in terms of spectrogram reconstruction quality in the learned timbre domain and transfer quality from other sources.

Results

Comparative Model Evaluation. The test set reconstruction quality of the models is evaluated by comparing the spectrogram magnitudes of the input and output waveforms using the Log-Spectral Distance (LSD). The instrument timbre transfer accuracy is evaluated by auto-encoding every other instrument subsets from URMP and Phenicx (besides the trained target) and every singing excerpts from VocalSet and predicting the instrument label of the

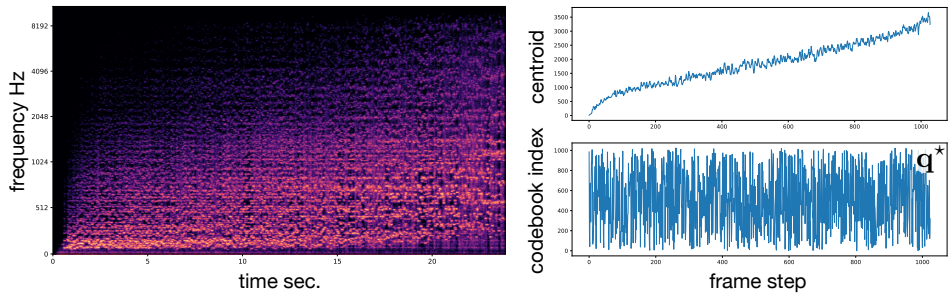

Figure 40: The spectrogram and centroid of an audio synthesized with an increasing centroid target in the violin representation. The corresponding series of embedding indexes $\mathbf{q}^\star$ does not exhibit any structure but can be arranged consistently with the acoustic descriptor target.

synthesized audio with the pretrained reference classifier. The accuracy is reported with respect to the target instrument, and aims to be maximized. In addition, the source f_0 and loudness curves are compared with those of the audio transfer. We use the Dynamic Time Warping (DTW) distance to measure how well the model preserves pitch and loudness independently from transferring timbre. The DTW score is normalized across audio excerpts by scaling the time series in unit range and averaging by the lengths of the DTW paths. For the model trained on singing voice, we transfer audio from all the instrument subsets and only report the average DTW distances.

The discrete representation of the VQ-VAE consistently improves the unsupervised timbre transfer accuracy in comparison with the baseline auto-encoder. For inference on other source domains, our proposed model solely uses a fixed basis of latent features learned from the spectral distribution of a given timbre. As a result, the quantization enforces audio transfer of the target timbre properties. We also observe that the disentangled gain prediction tends to improve the reconstruction of loudness, as shown by a lower average DTW distance for the VQ-VAE model. However, we did not constrain the model to rely on an explicit estimate of the fundamental frequency. Since it is not disentangled from the representation, we observe that quantization comes at the expense of a lesser accurate reconstruction of the pitch than for the continuous baseline model. Notably, in the VQ-VAE this property is bound to the trained instrument tessitura. The overall reconstruction quality in the target timbre, assessed with the test set LSD, is similar for both auto-encoders.

Besides the quantitative evaluation of the discrete representation against the baseline auto-encoder, we note two additional benefits of feature quantization. When processing out-of-domain audio of lower quality, such as vocal imitations recorded in uncontrolled conditions, the transfer ability is paired with denoising. Indeed, acoustically inconsistent features are discarded in the latent projection to a trained domain such as musical studio recordings. This

facilitates the use of timbre transfer from diverse recording environments such as for voice controlled synthesis. Moreover, we show that learning a discrete latent representation enables a direct mapping to acoustic descriptors as an other mean of high-level synthesis control.

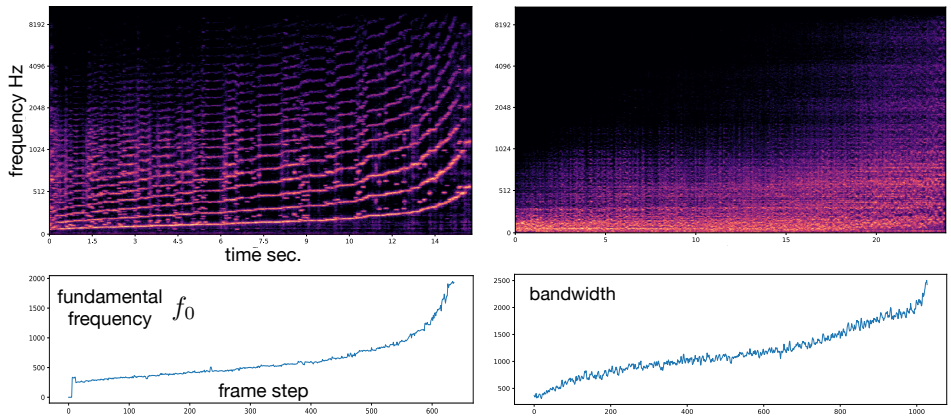

Figure 41: Controlling the sound synthesis of the violin by the f_0 (left) and the cello by the bandwidth (right).

Descriptor-Based Timbre Synthesis. In comparison with the baseline auto-encoder, the VQ-VAE decoder optimizes generation solely based on a discrete latent codebook. We introduce a mapping method for controllable sound synthesis (detailed in the supplementary material). Each embedding vector $\mathbf{q}^k$ with $k \in [1, K]$ approximately corresponds to a short-term timbre feature and a spectral filter $\tilde{\mathbf{H}}^k = D(\mathbf{q}^k)$. Given that the decoder has a RNN, some temporal relationships are introduced in the overlap-add subtractive synthesis. We decode a series of an individual feature $\{\mathbf{q}^k, \dots, \mathbf{q}^k\}$ and compute the average acoustic descriptor value $\tilde{a}^k$. After analyzing every latent vector, we obtain the mapping $\{\mathbf{q}^1, \dots, \mathbf{q}^K\} \leftrightarrow \{\tilde{a}^1, \dots, \tilde{a}^K\}$.

We can perform acoustic descriptor-based synthesis from a target a_i of any length M with $i \in [1, M]$ by selecting the nearest values in the discrete mapping $\tilde{a}_i^\star$ and decoding the corresponding series of latent features $\mathbf{q}_i^\star$. The mark * is used here to denote the nearest embedding elements to the descriptor target, whhich differs from the selection of $\mathbf{q}^*$ done by matching with the encoder output. Using such mapping, we show that we can control a VQ-VAE model of violin with an increasing *centroid* target. The decoded audio has a consistent spectrogram and synthesized centroid. We also observe that the acoustically ordered series of latent features corresponds to an unordered traversal of the discrete embedding. In other words, the index positions in the quantization space do not correlate to acoustic similarities, which are only provided by our proposed mapping method.

This analysis can be performed for other acoustic descriptors and other instrument rep-

resentations. We depict the control of the VQ-VAE model by a target defined either with *fundamental frequency* for the violin or with *bandwidth* for the cello. Our proposed model does not rely on f_0 conditioning in order to process diverse audio sources, such as vocal imitations without pitch. However, we show that the fundamental frequency can be controlled by mapping the unsupervised representation. Our proposed method yields an approximate decomposition of the acoustic properties of an individual timbre, it allows high-level and direct controls for sound synthesis.

Conclusion

We have introduced a raw waveform auto-encoder to learn a discrete representation of an individual timbre that is disentangled from loudness. It can be used for unsupervised transfer of musical instrument performances and singing voice. The model generates audio by subtractive sound synthesis, a technique which neither restricts the types of signals nor the duration that can be processed. The spectral distribution of a timbre is quantized with a set of short-term latent features that are decoded into noise filtering coefficients. This discrete representation can be mapped to acoustic properties in order to perform direct descriptor-based synthesis. Some descriptor targets can be matched with latent features that are decoded into signals with the desired auditory qualities. For instance, the unsupervised model can be controlled with the fundamental frequency. In addition, we experiment with transferring vocal imitations into an instrument timbre as an example of voice-controlled sound synthesis.

Interfaces for timbre manipulations. As discussed throughout this thesis, pitch and velocity have a well defined sense in music production whereas the remaining variations in timbre do not adhere to any commonly established representation besides that of instrument classes and categories of playing techniques. Digital synthesizers allow to continuously modulate the synthesis of timbre through hand-crafted signal processing pipelines, however the interface design remains an open question [235] in order to provide parameters that align with properties of acoustic perception and music semantic. We approached this problematic by invertible representation learning with auto-encoders, such that the dimensions of the latent representation would allow visualisation (analysis) and synthesis control with the aim to facilitate interactions using semantic regularisations, hierarchical and conditional information structures as well as learned feature quantisation. This raises multiple challenges, including the interpretability of the generative representations as well as the integration of these models in a common user interface that would allow switching between training datasets and neural audio synthesis engines while maintaining a generic interaction layout. To this end, a MaxMSP interface was developed and populated with models created by the ACIDS team at IRCAM[22] (Figure 42). The interface allows analysis and synthesis with several interactions to manipulate the audio output by controlling latent dimensions, conditioning variables and interpolating between multiple inputs. As the latent series remain highly dimensional, we use PCA reduction into a common number of salient orthogonal dimensions that are ranked and exposed to the user for creative exploration and empirical fine-tuning based on the rendered output variations.

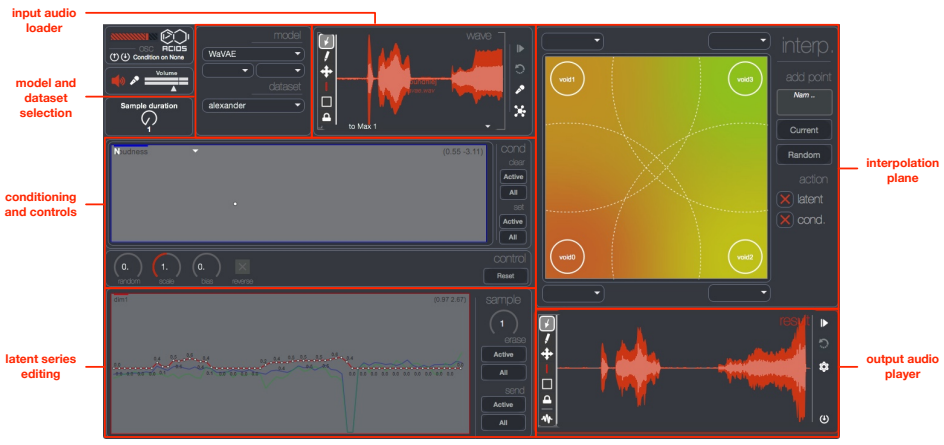

Figure 42: The ACIDS interface for timbre latent space exploration.

[22] https://acids.ircam.fr

The ACIDS interface is designed for continuous latent models, amongst which was integrated the unquantised version of the neural subtractive synthesis model detailed previously. Besides timbre transfer, we propose another generative application for the vector-quantization timbre model that takes advantage of the fixed number of learned latent features. In the first place, we are interested in visualising the distribution of these unsupervised short-term features with respect to classical acoustic descriptors. To do so, we individually decode audio from each of the quantisation vectors so that we can compute the corresponding acoustic quantities such as the spectral centroid. We use this analysis as a mapping function for descriptor-based synthesis (Figure 43) given a user provided acoustic target. Accordingly, we can traverse the latent codebook in the increasing order of the analysed descriptor and validate the acoustic behaviour of the synthesised audio. We extend this process to user provided targets by mapping each point of the descriptor curve with its nearest codebook element given the individual acoustic quantities that have been analysed. The resulting series of quantisation vectors is decoded into an audio that follows the desired acoustic variation. This work was submitted to the second international conference on timbre, held virtually and presented as an online poster. Some additional visualisations, audio samples and videos are hosted on the dedicated online repository: `https://acids-ircam.github.io/timbre_exploration/`.

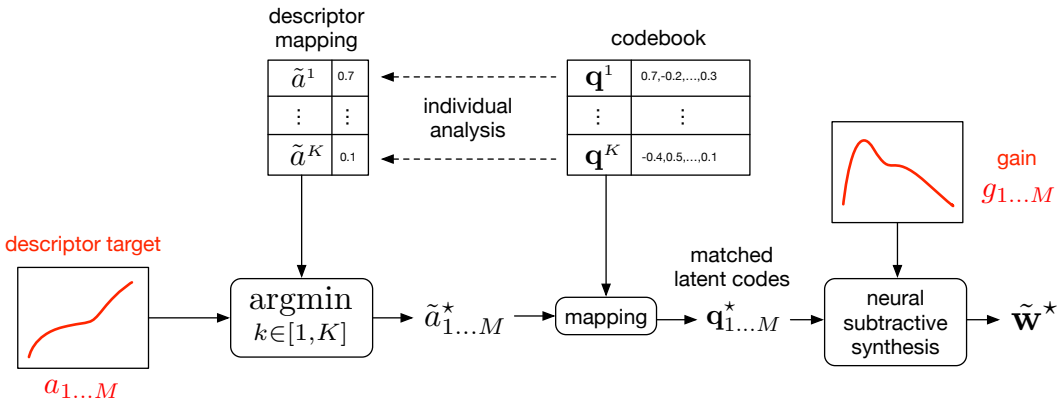

Figure 43: Descriptor-based synthesis by analysis and mapping of the learned vector-quantization codebook with acoustic descriptors, red elements denote the user controls. Each latent vector $\mathbf{q}^k$ is individually decoded and analysed with a scalar acoustic descriptor $a(.)$ in order to obtain paired tables of values $\{\mathbf{q}^1, \dots, \mathbf{q}^K\} \leftrightarrow \{\tilde{a}^1, \dots, \tilde{a}^K\}$. Given a user provided descriptor target $\{a_1, \dots, a_M\}$, the algorithm selects the nearest analysed values $\{\tilde{a}^\star_1, \dots, \tilde{a}^\star_M\}$ which are mapped to the corresponding codebook vectors $\{\mathbf{q}^\star_1, \dots, \mathbf{q}^\star_M\}$ and decoded along with a user specified gain envelope $\{g_1, \dots, g_M\}$.

Timbre Latent Space: Exploration and creative aspects

Antoine Caillon, Adrien Bitton [23], Brice Gatinet & Philippe Esling

(original publishing template of the 2nd International Conference on Timbre available at `https://arxiv.org/abs/2008.01370`)

Introduction

Recent studies show the ability of unsupervised models to learn invertible audio representations using Auto- Encoders [71]. While they allow high quality sound synthesis and high-level representation learning, the dimensionality of the latent space and the lack of interpretability of each dimension preclude their intuitive use. The emergence of disentangled representations was studied in Variational Auto-Encoders (VAEs) [145][111] and has been applied to audio. Using an additional perceptual regularization [72] can align such latent representation with the previously established multi-dimensional timbre spaces, while allowing continuous inference and synthesis. Alternatively, some specific sound attributes can be learned as control variables [17] while unsupervised dimensions account for the remaining features. In this paper, we propose two models and suited interfaces that were developed in collaboration with music composers in order to explore the potential of VAEs for creative sound manipulations. Besides sharing a common analysis and synthesis structure, one has a continuous latent representation and another has a discrete representation, which are applied to learning and controlling loudness invariant sound features.

Models

We consider a dataset of audio samples, such as performance recordings of an instrument. A variable-length audio $\mathbf{x}$ can be processed by analyzing series $\{\mathbf{x}_0, \dots, \mathbf{x}_L\}$ of signal windows $\mathbf{x}_i \in \mathbb{R}^{d_x}$ with an encoder E_ϕ mapping each frame into a latent code as $E_\phi : \mathbf{x}_i \rightarrow \mathbf{z}_i \in \mathbb{R}^{d_z}$. This encoder is paired with a decoder D_θ that inverts these features as $D_\theta : \mathbf{z}_i \rightarrow \hat{\mathbf{x}}_i$. The vanilla auto-encoder optimizes its parameters $\{\theta, \phi\}$ on a reconstruction objective such that $\hat{\mathbf{x}}_i \approx \mathbf{x}_i$.

Usually, we choose $d_z \ll d_x$ so that the latent variables embed a compressed representation of the data from which we can synthesize new samples. However, this continuous

[23] equal contributions, with Antoine's work on the continuous model

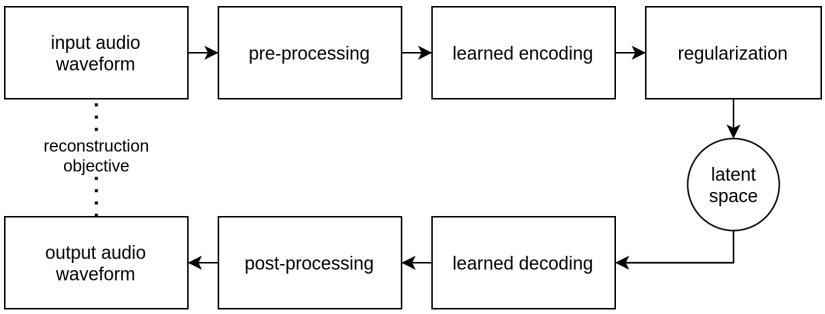

Figure 44: Block diagram of a VAE with optional pre and post audio processing.

representation often remains highly- dimensional and does not disentangle data properties on separate latent dimensions. The usability of such representation and its quality for sampling or interpolation are thus limited. These considerations highlight the need for additional training objectives that enforce useful properties in the latent representation. We consider two separate models, comparable in their overall encoder-decoder structure, but different in how the representation is regularized during training.

Continuous model. The first model aims to construct a latent space that is invariant to loudness in order to embed features that mainly account for the instrument timbre. It is achieved with an adversarial domain adaptation, where a latent regressor is trained at predicting loudness, and a gradient reversal optimization [85] leads to a loudness-invariant encoder representation. Besides this objective, the VAE latent space is regularized on a Gaussian prior distribution $\mathcal{N}(0, 1)$ which ensures local smoothness and favors independence between latent variables.

Discrete model. The second model is based on the Vector-Quantized VAE (VQ-VAE) proposed in [270]. It optimizes a discrete set of latent features $\mathbf{q}^j$. Each encoder output is matched to its nearest codebook element $\mathbf{q}_i^* \in \{\mathbf{q}^0, \ldots, \mathbf{q}^K\}$, before being decoded. This latent space is disentangled from a gain applied to the decoder output, which produces short-term features that are invariant to audio levels. Given that the set of latent features $\mathbf{q}^j$ is finite, we can analyze and map this codebook with acoustic descriptors.

Both models are intended to learn latent audio features that are invariant to loudness. The continuous model offers unconstrained and smooth feature manipulations. The discrete model can be analyzed in order to predict the output acoustic features embedded in the representation.

Experiments

Descriptor-based synthesis. Each vector of the discrete representation is individually decoded and the output signal is analyzed with a descriptor. It is thus possible to compute the mapping between a descriptor curve and the series of nearest latent features (details in [18]). Latent synthesis can be directly controlled by following a user-defined descriptor target, as shown in the following figure. The codebook can be ordered and traversed according to different properties, such as centroid or fundamental frequency.

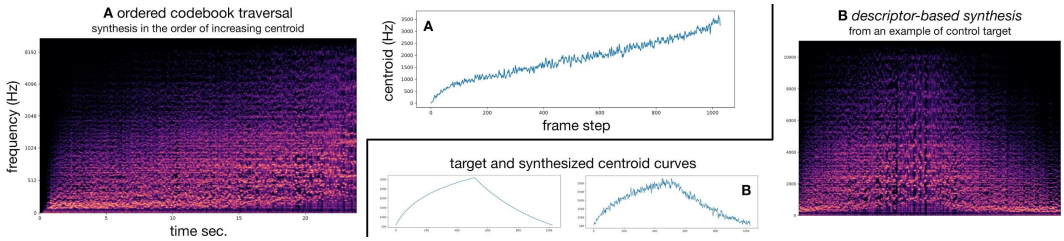

Figure 45: The discrete representation can be analyzed with the spectral centroid and traversed in the increasing order (A). A control target can be synthesized by selecting the nearest latent features, the decoded audio approximately follows the curve provided (B).

Continuous latent interpolations. In order to display the local smoothness of the continuous model, we consider the time variant linear interpolation $\mathbf{z}_{\text{interp}}$ between two latent series $\mathbf{z}_a$ and $\mathbf{z}_b$ of the same size inferred from two audio samples A and B. Decoding $\mathbf{z}_{\text{interp}}$ results in an audio sample smoothly interpolating between sample A and sample B (last figure). In order to facilitate a creative use of this model, we present two interfaces designed to circumvent the problem of identifying latent dimensions by facilitating their exploration.

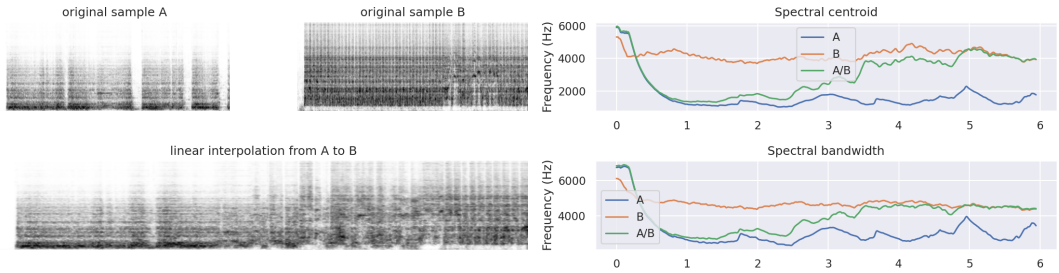

Figure 46: Linear interpolation in the latent space between two audio samples. We can see that the centroid and the bandwidth of the interpolated audio sample performs a smooth transition between those of the two original audios.

Continuous model interfaces

The first interface is a Max/MSP application that is a graphical equivalent to the command line tools we usually have to test the model. It features several high-level interactors such as mathematical operators on the latent series, manual editing, and an interpolation plane. We have built this application in collaboration with A. Schubert [24], aligning with his remarks on how to improve visualization and control over the generation. This interface is intended to be used in order to grasp the main characteristics of a model trained on a specific dataset.

This stand-alone interface has built-in interactions but a limited integration and restrictions in the possible operations. We have thus developed a second interface built in collaboration with B. Gatinet, implementing the encoder and the decoder as PureData abstractions that can be combined with any other regular objects. New aspects of the continuous model emerge from this interface, as it allows uninterrupted exploration with realtime rendering, enabling the use of complex signal processing techniques on both the audio and latent series. As this interface can be integrated in real time inside a digital audio workstation, it is more suited for composition workflows. It is furthermore a strict superset of the first interface in terms of functionalities.

The use of these interfaces has brought to light new ways of generating audio signals, whether by explicit control of an audio descriptor, or by morphing between different existing sounds. Training a model on an audio domain and using it to resynthesize an audio sample from a different domain can also lead to an implicit synthesis method. Additional results on audio conversion of instrument sounds can be found in [18].

Conclusion

This research has studied VAEs with continuous and discrete latent sound representations as creative tools to explore timbre synthesis. The discrete model allows the generation of a new audio signal by directly controlling acoustic descriptors. Manipulations of the continuous model are eased by developing specific interfaces and real-time rendering, which greatly enrich composition and sound design possibilities. And in turn, it gives further insights on the generative qualities found in the learned representations, as well as the relevance of their different parameters and controls with respect to the new timbres that are synthesized.

---

[24] See `http://www.alexanderschubert.net/`

5.3 Light-weight Neural Audio Processing

Besides the interpretability of the learned representations and the intuitivity of user interactions, neural audio synthesis models face the challenge of efficiency. This mainly refers to the computational resources required to train and deploy models, as well as the amount of data that needs to be collected which is often dependent on the model complexity. As introduced in section 3.3.5, the early works which aimed at reducing the complexity of trained models were mainly based on masking the trained weights of lowest magnitude (i.e. zeroing their activations) and fine-tuning the kept weights. The pruning and fine-tuning approaches have remained little satisfying due to the drop in performance and the limited ratio of weights that could be masked. A recent breakthrough in model pruning was brought by the lottery ticket hypothesis [80] which postulates that a randomly initialised network (i.e. before training) already contains its efficient sparse sub-networks which are called wining tickets. This model compression algorithm works by iteratively identifying the trained wining tickets (e.g. largest magnitude), rewinding to the random initialisation state, masking and re-training in isolation the selected weights until convergence. After several pruning and re-training steps it is shown that the sparsified models can outperform the original dense model with unprecedented masking ratios of more than 90%.

In order to effectively reduce the model size (i.e. memory requirement) and number of operations for inference (i.e. computation requirement) of sparsified models, the lottery ticket hypothesis is implemented with weight trimming (Figure 47) and applied to audio synthesis [74] and music information retrieval [73] in a study led by Pr. Esling. While masking can be applied in an unstructured manner, trimming requires removing entire structural units (e.g. channels of a convolution) in order to reshape the sparsified model into its smaller counterpart that is rewound and trained in isolation according to the algorithm of the lottery ticket hypothesis. The results across multiple audio processing tasks and architectures, including discriminative and generative ones, show that for trimming up to 50% the model performance is consistently improved and that high trimming ratios above 90% can be achieved at the expense of moderate drops in performance. The study is carried on MIR tasks such as classification, pitch estimation and transcription as well as on neural audio synthesis models (WaveNet [268], DDSP [70], SING [52]) that are trimmed according to several weight ranking criteria. The original ranking is based on a magnitude criterion computed on an entire unit, it is

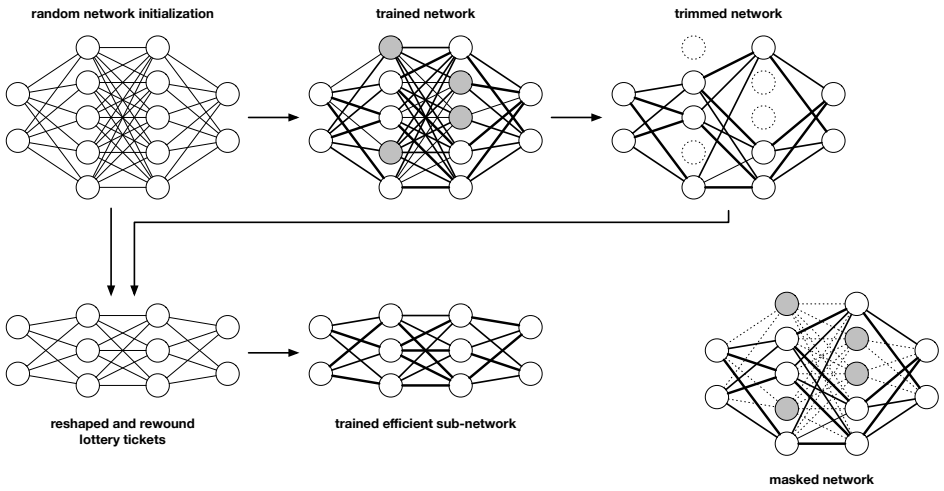

Figure 47: Model compression by weight trimming under the lottery ticket hypothesis. After training, the weights are ranked and removed before reshaping the kept units and rewinding them to their initial random states. As a result the trained sub-network is effectively reduced in size and number of operations, as opposed to the sparsification by masking (bottom right). For simplicity we only show a single step of trimming, in practice the model compression is progressively done in several iterations of training, trimming and rewinding. The figure is adapted after [73].

compared with a criterion on the summed unit activations obtained by cumulative forward pass of the whole training dataset or a criterion based on the learned scaling parameters of the layer batch-normalisation (if applicable). It appears that across these tasks, models and datasets, the cumulative activation is the most stable criterion for weight selection and trimming.

This study is of high-relevance for the general deep learning research as trimming can drastically reduce the energy consumption of a model throughout its life-long deployment. Moreover, it calls for a more critical approach to model evaluation as it seems that the race for over-parametrisation could actually hinder greater leaps in performance. Specifically to the field of audio, model compression seems to be one of the most realistic solutions for real-time processing and integration within usual hardwares (e.g. laptops, mobile phones, embedded devices) which have been major challenges in the dissemination of neural audio synthesis models as tools for broader user communities and musicians.

6 Conclusion

6.1 List of publications

Here is recapitulated in chronological order the list of authored and co-authored academic publications during the thesis.

2018

- ▷ Adrien Bitton, Axel Chemla-Romeu-Santos and Philippe Esling - Timbre transfer between orchestral instruments with semi-supervised learning - First International Conference on Timbre[25], Montreal, Canada.
- ▷ Axel Chemla-Romeu-Santos, Adrien Bitton, Goffredo Haus and Philippe Esling - Unsupervised timbre spaces through perceptually-regularized variational learning - First International Conference on Timbre, Montreal, Canada.
- ▷ Philippe Esling, Axel Chemla-Romeu-Santos and Adrien Bitton - Generative timbre spaces: regularizing variational auto-encoders with perceptual metrics - International Conference on Digital Audio Effects (DAFx-18), Aveiro, Portugal.
- ▷ Philippe Esling, Axel Chemla-Romeu-Santos and Adrien Bitton - Bridging audio analysis, perception and synthesis with perceptually-regularized variational timbre spaces - International Society for Music Information Retrieval Conference (ISMIR-18), Paris, France.
- ▷ Adrien Bitton, Philippe Esling and Axel Chemla-Romeu-Santos - Modulated variational auto-encoders for many-to-many musical timbre transfer - Arxiv 1810.00222.

2019

- ▷ Adrien Bitton, Philippe Esling, Antoine Caillon and Martin Fouilleul - Assisted sound sample generation with musical conditioning in adversarial auto-encoders - International Conference on Digital Audio Effects (DAFx-19), Birmingham, United Kingdom.

[25] `https://www.mcgill.ca/timbre2018/files/timbre2018/timbre2018_proceedings.pdf`

2020

- ▷ Adrien Bitton, Philippe Esling and Tatsuya Harada - Neural granular sound synthesis - International Computer Music Conference (ICMC), accepted in 2020, conference postponed to 2021[26].

- ▷ Adrien Bitton, Philippe Esling and Tatsuya Harada - Vector-Quantized timbre representation - Arxiv 2007.06349.

- ▷ Antoine Caillon, Adrien Bitton, Brice Gatinet and Philippe Esling - Timbre latent space: exploration and creative aspects - Second International Conference on Timbre[27], Thessaloniki, Greece.

- ▷ Philippe Esling, Ninon Devis, Adrien Bitton, Antoine Caillon, Axel Chemla-Romeu-Santos and Constance Douwes - Diet deep generative audio models with structured lottery - International Conference on Digital Audio Effects (DAFx-20), Vienna, Austria.

- ▷ Philippe Esling, Théis Bazin, Adrien Bitton, Tristan J. J. Carsault and Ninon Devis - Ultra-light deep MIR by trimming lottery tickets - International Society for Music Information Retrieval Conference (ISMIR-20), Montreal, Canada.

- ▷ Hayato Sumino, Adrien Bitton, Lisa Kawai, Philippe Esling and Tatsuya Harada - Automatic music transcription and instrument transposition with differentiable rendering - The Joint Conference on AI Music Creativity, Stockholm, Sweden.

6.2 Related projects

Besides the experiments carried and published during this thesis, several projects have been done at the intersection of applied machine learning research and music. These unique opportunities allowed to combine academic experience with practical skills, tackle novel problems and interact within broader communities.

In 2018, I applied and was given the opportunity to participate in a week long

[26] `http://icmc2021.org/selected-papers/`
[27] `http://timbre2020.mus.auth.gr/assets/papers/Timbre2020_proceedings.pdf`

residency organized by Club Transmediale [28] in Berlin around the topic of artificial intelligence and music which led to a public performance [29] at the Hebbel am Ufer theater. Our piece "The man who never told a lie" questioned the role of the human performer in the realm of data-driven content generation with machine learning tools. In that sense, the piece mixed the live audio and video recordings with artificially processed audio samples and images sampled in real-time from the performance. In order to blur the boundary between true and synthetic contents, fake gestures were displayed to hinder the audience discernment. As a result, this improvisation creates an ambiguous discourse between the performer and the content generation which collaborate (e.g. real-time sampling and answering) into confusing and surprising the audience judgement.

In 2020, I had the chance to co-organise with Professor Suguru Goto a workshop on artificial intelligence and music [30] at The Tokyo University of the Arts (Geidai) which was an opportunity to gather both composers and scientists to share research works and knowledge across our related communities. The program equally featured presentations with art and science backgrounds, leading to fruitful debates towards practical applications of machine learning tools for composers and reflections on the development of these technologies. This project took place during a seven-month visit at the Machine Intelligence Laboratory of The University of Tokyo (Todai) thanks to the support of Professor Tatsuya Harada and a Japan Society for the Promotion of Science (JSPS) short-term fellowship.

In 2020, I applied and was given the opportunity to participate in the hackathon on artificial intelligence and music organized virtually by ARS Electronica [31]. I took part in the team working on the topic of "Designing user interactions when Humans and Machine Learning models are together in the musical loop" which was supervised by Lamtharn Hanoi Hantrakul. In the course of this week, our team developed a generative music experience for virtual reality [32]. This project called "Exploring Memories" lets the audience navigates in a 3D point cloud landscape which was mapped to the latent space of a pre-trained melody generation model and rendered with real-time spatial audio. By placing objects in this landscape, the audience can discover the melodic

[28] https://archive2013-2020.ctm-festival.de/archive/festival-editions/ctm-2018-turmoil/transfer/musicmakers-hacklab/

[29] https://youtu.be/Tv48dC48UvE?t=817

[30] https://adrienchaton.github.io/seminar_geidai_AI_Music/

[31] https://ars.electronica.art/keplersgardens/en/aixmusic-hackathon/

[32] https://youtu.be/G2jLPT72ko8?t=969

structures learned by the model (e.g. various note densities) and create a soundscape which unveils the neural network memory.

6.3 Conclusion and future works

In this thesis, we have focused on the acoustic production of the musical timbre which entails the perceptual qualities of different instrument sounds by which a given composition may be rendered in the audio domain. While a common notation system exists to define music in the symbolic domain, the representations and interactions in the acoustic domain have remained more elusive. This shortcoming could be understood by identifying the following limitations. Textual metadata that usually structure audio sample libraries can only express coarse categories of sounds but do not describe their subjective values nor their continuous variations and relationships which play a central role in music production. Analysis spaces built from psychoacoustic studies and listening tests can represent these relationships but cannot generalise to unrated samples nor they can be inverted to synthesise new sounds. Digital signal processing techniques have enabled the development of a great variety of synthesizers which allow continuous controls for generating diverse timbres. Yet, these systems feature many parameters with complex non-linear relationships which require a tedious exploration of parameter configurations. Amongst these techniques, analysis and synthesis models are of great interest as one may extract parameters from some given samples, visualise and manipulate them to render new audio variations. Nonetheless, classical sound models are mostly bound to express low-level signal properties and their representations remain hard to visualise at a large scale and to manipulate with respect to some target perceptual properties. To this end, we adopted a data-driven approach for large scale analysis and synthesis with auto-encoders in order to learn more meaningful representations and interactions grounded in the acoustic domain. In this probabilistic setting, the whole library of audio samples is treated as an empirical observation sampled from the underlying distribution of perceived sounds we aim to model. Throughout these thesis experiments, we have applied and refined several machine learning approaches for audio density estimation which share some common strengths and challenges. Data-driven models are inherently designed to process data at large scales, by doing so they can learn underlying relationships of sounds which are the basis of understanding higher-level data properties. Deep neural network optimization offers a great flexibility in

estimating these densities which cannot be evaluated in closed-form while enabling to define additional modelling constraints by which we aim to structure useful generative representations. On the other hand, applications of machine learning tools to musical sound synthesis must achieve an audio fidelity comparable to signal processing techniques while maintaining a commensurate computational cost for integration in the usual hardware and dissemination to broad audiences. Since the audio signal is sampled at a very high frequency, an ongoing challenge is that of efficiently modelling both local properties and longer-term relationships which span an humongous number of dimensions as we deal with music structures in the order of seconds to minutes. Finally, the evaluation of these generative models is another ongoing topic as their optimization objectives are often not explicitly traducing the perceptual properties of the data nor they directly assess the creative potential of the learned models besides the training simulation. Accordingly, we recapitulate our different contributions and draw directions for future works.

Timbre processing by domain translations. In this first approach, the processing of timbre was posed as the conversion between two or more libraries of audio samples which define different perceptual categories. We referred to this technique as an implicit timbre modelling as we did not explicitly represent the features that describe each of these domains, nor we annotated which transformation should be applied to a given sample so that it is translated into the target timbre domain. In this setting, without ground-truth pairs of samples across domains, we learned such mapping through specific training objectives which aim at preserving a domain invariant content while altering the domain specific features to match that of the target. One major challenge is that of scaling to translations across many domains, mostly because the domain specific mapping is usually enforced by individual adversarial classifiers in each data domain. To this end we proposed to use non-parametric kernel distances as the basis of learning the domain specific distributions, which rely on the two-sample test approach rather than on the adversarial approach. As we did not need anymore to increase the number of discriminators as we increase the number of domains, we used domain conditioning which allows their processing into a single auto-encoder model. This implicit timbre processing results in a synthesis driven by example, in that sense the user can provide a sample belonging from one source domain and let the model operate the timbre transfer to the specified target. As we may want to control the pitch transformation apart from

the timbre, we as well relied on semi-supervision by conditioning the model with the semitone and octave classes. We as well proposed to visualise the distributions of acoustic descriptors in each domain against that of the translated samples. While the model was not provided any supervision with respect to these features, this evaluation shows that synthetic samples mapped in the target domains do follow the observed acoustic distributions.

Timbre representations by conditioning. In order to allow more expressive controls on the audio synthesis, we proposed to learn sampling with conditioning on finer musical attributes of timbre such as playing styles or any other subjective labels. These non-invertible features are only partially describing the target timbre, thus we complemented them with an analysis and synthesis representation that captures the remaining acoustic variations belonging to a given set of attribute targets. One challenge in this setting is that the unsupervised features extracted by the encoder may embed those which we wish to condition on, thus the decoder can freely bypass the attribute targets that ultimately cannot be controlled during sampling. In order to prevent the pathway memorisation between the encoder features and the decoder reconstruction, we applied an adversarial regularisation in the latent space which pushes the representation to be invariant to the target attributes. As the encoder features should not allow classification of these attributes, the decoder must effectively use the conditioning information which subsequently enables a controlled sampling. This was evaluated by pre-training data classifiers on the desired musical attributes and assessing the accuracy of the generated samples with respect to the provided conditioning targets. In the aforementioned experiments the models were trained on magnitude spectrograms, rather than the raw waveform, which provide a more compact and structured acoustic representation but require some approximate phase estimation for audio synthesis. This is commonly done by Griffin-Lim iterations which cause some latency and a limited audio fidelity. Alternatively, we adopted the feed-forward neural inversion by training a spectrogram vocoder model which can directly synthesise waveform at the output of our conditional spectrogram generator.

Hierarchical timbre representations. Learning musical audio representations is an ongoing challenge which involves modelling both local acoustic properties which ensure a high-quality signal generation and longer-term relationships at multiple scales

corresponding to increasing musical contexts. This may be facilitated by using short-term acoustic representations such as magnitude spectrograms, although it prevents from end-to-end learning for direct waveform synthesis. When combining a spectrogram generation model with a phase approximation model, usually trained separately, we encounter error accumulations and only learn lossy acoustic representations. Inspired by granular synthesis techniques, we proposed a hierarchical model which learns a lower-level representation of short waveform windows which are individual audio grains organised by acoustic similarity. On top of this granular representation, a recurrent embedding jointly learns the temporal structures of series of grain features to form audio events such as a note with a vibrato or a drum hit with an attack and release. We introduced a method for efficient waveform synthesis by subtractive noise filtering for every signal windows that are assembled by overlap-add into the output signal and refined with a learned post-processing module. We show that this model improves both the reconstruction quality and synthesis speed in comparison with commonly used up-sampling convolutions and that it is flexible enough to model both pitched sounds, drum kit sounds and environmental noises. As the temporal structure is learned on a down-sampled granular representation, the hierarchical architecture only requires a shallow recurrent embedding. By performing interpolations in the lower-level grain latent space, we can continuously generate smooth acoustic textures of variable lengths and extend some of the classical granular synthesis techniques. By sampling the temporal embedding, we can generate fixed-length audio events and create a spectro-temporal morphing between features of two samples.

Discrete timbre representations. Another approach to timbre transfer was proposed which relies on learning a discrete representation of a single timbre that is decomposed into a finite number of learned latent features. For this experiment we took advantage of the aforementioned short-term waveform processing by noise filtering and overlap-add, which we train with a gain envelope that is predicted in addition to latent features and used to scale the output filters. As a result, the discrete representation does not need to embed the amplitude information but the fundamental frequency and spectral distribution of the target timbre. By encoding and quantisation of an audio from another source, we can perform timbre transfer which amounts to reconstruction with the discrete features learned in the target timbre. In this setting, we do not need to train on these other domains and can thus convert from unseen sources including

vocal imitations as a mean of voice-driven synthesis. Moreover this process can be applied to variable-length audio, as opposed to the usual domain translation techniques. As the learned discrete representation decomposes the target timbre into a finite set of features, we proposed a novel method to analyse and control synthesis from the latent space using acoustic descriptors. By individually decoding each latent feature and computing a given acoustic descriptor, we can visualise the distribution of the discrete latent space and map a user-specified descriptor envelope to the series of nearest latent features that are decoded into an audio that follows the provided target. This mapping allows a direct control for descriptor-based synthesis.

Usability of neural audio processing and synthesis models. In the first place, we described our experiments with an emphasis on learning strategies, neural network architectures and analysis/synthesis representations for manipulating timbre and generating musical audio. We as well proposed some evaluation methods to assess the effectiveness of our proposed models in terms of both reconstruction quality and accuracy of the learned controls. Yet the usability of these tools for music production greatly depends on two factors which are the interactions and interface design for the user as well as the efficiency with respect to computational requirements and the ability to learn from datasets of moderate sizes. In addition to the aforementioned interactions such as voice-driven and descriptor-based synthesis, we have worked on some prototype interfaces which can be run within general music software environments such as Max/MSP. A drum machine was implemented to run a model conditioned on generating eight different classes of drum hits that are triggered with a step sequencer. The audio buffers that are played-back can be randomly sampled with low latency to explore many variations pertaining to a given drum class. Besides explicit control by conditioning, an interface for the exploration of unsupervised latent spaces was developed to host several models of the ACIDS/IRCAM team. Their unsupervised dimensions can be treated as synthesis parameters, nonetheless the corresponding output variations are often not predictable and mostly randomly sampled or interpolated. To allow the direct exploration of these dimensions, the latent spaces are projected onto a common number of orthogonal dimensions by PCA and different visualisations and operations can be applied to the encoding of several samples for intuitively generating new variations. Lastly, the computational efficiency of neural audio processing models could be dramatically improved on a large array of tasks including both information retrieval and

synthesis. In this study we observe that most trained models are over-parameterised and only a fraction of these parameters ultimately contribute to their accuracy. By identifying them, trimming unnecessary parameters and retraining the most efficient sub-networks we can maintain the target performance with down to 10% of the architecture capacity. This compression of trained models is a significant step towards light-weight and real-time applications that can run on general public hardware such as laptops or embedded devices, as well as a potential solution to deploying machine learning tools with a reduced energy consumption and environmental footprint.

Future works. In the years spanned by this thesis, we could witness great advances in neural audio synthesis amongst which the recently published Differentiable Digital Signal Processing (DDSP) is one of the most promising approach to musical audio synthesis. This model can generate high-quality audio in a few hours of training on datasets as small as tens of minutes. This is achieved within a modular architecture that incorporates elements of classical digital signal processing to efficiently model different components of sound perception such as harmonic partials, stochastic residuals and reverberation. The DDSP framework opens many exciting future directions of research which comprise the development of new modules that can fit more diverse sounds (e.g. non-harmonic, non-pitched and percussive audio), which was partially investigated within our proposed noise filtering and overlap-add synthesis approach. Yet much more techniques may be derived in the DDSP framework by integrating other descriptors than fundamental frequency and loudness as well as implementing processes inspired from synthesizer circuits or physical modelling as part of the synthesis engines. Because the DDSP model can be trained on a short audio duration, it can capture a very specific timbre and acoustic environment such as the sound of given violinist in a given recording. Another direction of future works is to integrate the DDSP synthesis as part of a modular score rendering pipeline, which has already received a certain attention [134]. In this setting, which is the topic of a current master's degree internship in co-supervision, the task is split between a control model that analyzes the quantised score information and generates expressive acoustic envelopes that are rendered to audio by the subsequent DDSP synthesizer. Such modular approach for score to audio is promising in many ways as we can expect much lighter-weight systems than those found in the earlier literature. Moreover, as the control envelopes processed by the synthesizer are explicitly traducing the instrument and musician dependent playing

style we can expect learning much more expressive models which could capture the whole performance style of given recording, on both acoustic and interpretation levels.

References

[1] S. Adiga, M. Attia, Wei-Ting Chang, and R. Tandon. On the tradeoff between mode collapse and sample quality in generative adversarial networks. *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1184–1188, 2018.

[2] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *J. Mach. Learn. Res.*, 15(1):3563–3593, January 2014.

[3] Aziz Alotaibi. Deep generative adversarial networks for image-to-image translation: A review. *Symmetry*, 12(10), 2020.

[4] Matthew Amodio and Smita Krishnaswamy. Travelgan: Image-to-image translation by transformation vector learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, July 2020. arXiv:2007.03051.

[6] Joseph M. Antognini, Matt Hoffman, and Ron J. Weiss. Synthesizing diverse, high-quality audio textures. *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 2018.

[7] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 609–617, 2017.

[8] Devansh Arpit, Yingbo Zhou, Hung Ngo, and Venu Govindaraju. Why regularized auto-encoders learn sparse representation? In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 136–144, New York, New York, USA, 20–22 Jun 2016. PMLR.

[9] S. Ö. Arık, H. Jun, and G. Diamos. Fast spectrogram inversion using multi-head convolutional neural networks. *IEEE Signal Processing Letters*, 26(1):94–98, 2019.

[10] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 892–900, Red Hook, NY, USA, 2016. Curran Associates Inc.

[11] P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. Velasco. Theory, implementation and applications of nonstationary gabor frames. *Journal of Computational and Applied Mathematics*, 236(6):1481–1496, 2011.

[12] Guillaume Ballet, Riccardo Borghesi, Peter Hoffmann, and Fabien lévy. Studio Online 3.0: An Internet "Killer Application" for Remote Access to IRCAM Sounds and Processing tools. In *Journées d'Informatique Musicale*, Paris, France, May 1999.

[13] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[14] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

[15] David Berthelot*, Colin Raffel*, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *International Conference on Learning Representations*, 2019.

[16] Stefan Bilbao, Brian Hamilton, Alberto Torin, Craig Webb, Paul Graham, Alan Gray, Kostas Kavoussanakis, and James Perry. Large scale physical modeling sound synthesis. In *Proceedings of the Stockholm Musical Acoustics Conference/Sound and Music Computing Conference*, August 2013.

[17] adrien bitton, Philippe Esling, Antoine Caillon, and Martin Fouilleul. Assisted sound sample generation with musical conditioning in adversarial auto-encoders. *Proceedings of the 22 nd International Conference on Digital Audio Effects (DAFx-19)*, September 2019.

[18] Adrien Bitton, Philippe Esling, and Tatsuya Harada. Vector-quantized timbre representation, 2020.

[19] Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C. Cobo, and Karen Simonyan. High fidelity speech synthesis with adversarial networks. In *International Conference on Learning Representations*, 2020.

[20] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the State of Neural Network Pruning? *arXiv e-prints*, page arXiv:2003.03033, March 2020.

[21] jonathan boilard, philippe gournay, and roch lefebvre. a literature review of wavenet: theory, application, and optimization. *journal of the audio engineering society*, march 2019.

[22] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41 – 65, 2019.

[23] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics.

[24] Jean-Pierre Briot. From Artificial Neural Networks to Deep Learning for Music Generation - History, Concepts and Trends. *Neural Computing and Applications*, October 2020. Special issue on Networks in art, sound and design, edited by Juan Romero and Penousal Machado.

[25] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[26] Judith C. Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[27] Christopher P. Burgess, Irina Higgins, Arka Pal, Loïc Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *CoRR*, abs/1804.03599, 2018.

[28] M. Caetano and X. Rodet. Musical instrument sound morphing guided by perceptually motivated features. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(8):1666–1675, 2013.

[29] Marcelo Caetano, George Kafentzis, Athanasios Mouchtaris, and Yannis Stylianou. Full-band quasi-harmonic analysis and synthesis of musical instrument sounds with adaptive sinusoids. *Applied Sciences*, 6:127, 05 2016.

[30] Lee Callender, Curtis Hawthorne, and Jesse Engel. Improving perceptual quality of drum transcription with the expanded groove midi dataset, 2020.

[31] Dermot Campbell, Edward Jones, and Martin Glavin. Audio quality assessment techniques—a review, and recent developments. *Signal Processing*, 89(8):1489 – 1500, 2009.

[32] Mark Cartwright and Bryan Pardo. Vocalsketch: Vocally imitating audio concepts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 43–46, New York, NY, USA, 2015. Association for Computing Machinery.

[33] Rodrigo Castellon, Chris Donahue, and Percy Liang. Towards realistic midi instrument synthesizers. 2020.

[34] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021.

[35] Nutan Chen, Alexej Klushyn, Richard Kurle, Xueyan Jiang, Justin Bayer, and Patrick Smagt. Metrics for deep generative models. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1540–1550. PMLR, 09–11 Apr 2018.

[36] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans. nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks. *IEEE Access*, 8:161981–162003, 2020.

[37] J. Chien and C. Wang. Variational and hierarchical recurrent autoencoder. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3202–3206, 2019.

[38] Keunwoo Choi and Kyunghyun Cho. Deep unsupervised drum transcription. In Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk, editors, *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, pages 183–191, 2019.

[39] Keunwoo Choi and Kyunghyun Cho. Deep unsupervised drum transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Delft, Netherland*, 2019.

[40] Keunwoo Choi, György Fazekas, Kyunghyun Cho, and M. Sandler. A tutorial on deep learning for music information retrieval. *ArXiv*, abs/1709.04396, 2017.

[41] Keunwoo Choi, György Fazekas, Mark B. Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. *CoRR*, abs/1703.09179, 2017.

[42] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.

[43] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[44] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2041–2053, 2019.

[45] John Chowning. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society*, pages J. Audio Eng. Soc. 21 (7), 526–534., 1973.

[46] Casey Chu, Kentaro Minami, and Kenji Fukumizu. Smoothness and stability in gans. In *International Conference on Learning Representations*, 2020.

[47] J. Cramer, H-.H. Wu, J. Salamon, and J. P. Bello. Look, listen and learn more: Design choices for deep audio embeddings. In *IEEE Int. Conf. on Acoustics,*

Speech and Signal Processing (ICASSP), pages 3852–3856, Brighton, UK, May 2019.

[48] A. Creswell and A. A. Bharath. Inverting the generator of a generative adversarial network. *IEEE Transactions on Neural Networks and Learning Systems*, 30(7):1967–1974, 2019.

[49] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[50] Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019.

[51] Shuqi Dai, Zheng Zhang, and Gus Xia. Music style transfer issues: A position paper. *arXiv preprint arXiv:1803.06841*, 2018.

[52] Alexandre Defossez, Neil Zeghidour, Nicolas Usunier, Leon Bottou, and Francis Bach. Sing: Symbol-to-instrument neural generator. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 9041–9051. Curran Associates, Inc., 2018.

[53] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[54] Sander Dieleman, Aäron van den Oord, and Karen Simonyan. The challenge of realistic music generation: Modelling raw audio at scale. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8000–8010, Red Hook, NY, USA, 2018. Curran Associates Inc.

[55] Xin Ding, Yongwei Wang, Zuheng Xu, William J Welch, and Z. Jane Wang. Cc{gan}: Continuous conditional generative adversarial networks for image generation. In *International Conference on Learning Representations*, 2021.

[56] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In Yoshua Bengio and Yann LeCun, editors, *3rd*

International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings, 2015.

[57] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[58] C.B. Do and S. Batzoglou. What is the expectation maximization algorithm? *Nature biotechnology*, 26(8):897–899, 2008.

[59] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2019.

[60] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *ICLR*, 2017.

[61] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 658–666. Curran Associates, Inc., 2016.

[62] Jonathan Driedger, Meinard Müller, and Sascha Disch. Extending harmonic-percussive separation of audio signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 611–616, Taipei, Taiwan, 2014.

[63] Jonathan Driedger, Thomas Prätzlich, and Meinard Müller. Let It Bee – towards NMF-inspired audio mosaicing. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 350–356, Málaga, Spain, 2015.

[64] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and C. Aaron Courville. Adversarially learned inference. *ICLR*, 2017.

[65] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 2018. https://distill.pub/2018/feature-wise-transformations.

[66] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *ICLR*, 2017.

[67] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, page 258–267, Arlington, Virginia, USA, 2015. AUAI Press.

[68] Taffeta M. Elliott, Liberty S. Hamilton, and Frédéric E. Theunissen. Acoustic structure of the five perceptual dimensions of timbre in orchestral instrument tones. *Acoustical Society of America Journal*, 133(1):389, January 2013.

[69] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. GANSynth: Adversarial neural audio synthesis. In *International Conference on Learning Representations*, 2019.

[70] Jesse Engel, Lamtharn (Hanoi) Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. In *International Conference on Learning Representations*, 2020.

[71] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1068–1077. JMLR.org, 2017.

[72] P. Esling, Axel Chemla-Romeu-Santos, and Adrien Bitton. Generative timbre spaces with variational audio synthesis. In *Proceedings of the International Conference on Digital Audio Effects*, 2018.

[73] Philippe Esling, Théis Bazin, Adrien Bitton, Tristan Carsault, and Ninon Devis. Ultra-light deep MIR by trimming lottery tickets. In *Proceedings of the 21th International Society for Music Information Retrieval Conference*, 2020.

[74] Philippe Esling, Ninon Devis, Adrien Bitton, Antoine Caillon, Axel Chemla-Romeu-Santos, and Constance Douwes. Diet deep generative audio models with structured lottery. In *Proceedings of the 23rd International Conference on Digital Audio Effects*, 2020.

[75] Giorgio Fabbro, Vladimir Golkov, Thomas Kemp, and Daniel Cremers. Speech synthesis and control using differentiable dsp, 2020.

[76] Otto Fabius, Joost R. van Amersfoort, and Diederik P. Kingma. Variational recurrent auto-encoders. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

[77] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, October 2016. Funding Information: The first author was supported by NSF grant DMS 1265524, AFOSR grant FA9550-12-1-0425 and U.S.-Israel Binational Science Foundation grant 2014055. The second author was supported by NSF grant EECS-1135843.

[78] Kelly Fitz and Lippold Haken. Lemur: A bandwidth-enhanced sinusoidal modeling system. *The Journal of the Acoustical Society of America*, 103(5):2756–2757, 1998.

[79] Derry FitzGerald. Harmonic/percussive separation using median filtering. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 246–253, Graz, Austria, 2010.

[80] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

[81] Brendan J. Frey and Geoffrey E. Hinton. Variational learning in nonlinear gaussian belief networks. *Neural Comput.*, 11(1):193–213, 1999.

[82] Jerome H. Friedman and Lawrence C. Rafsky. Graph-theoretic measures of multivariate association and prediction. *Ann. Statist.*, 11(2):377–391, 06 1983.

[83] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[84] M. Fröjd and Andrew Horner. Sound texture synthesis using an overlap-add/granular synthesis approach. *AES: Journal of the Audio Engineering Society*, 57:29–37, 01 2009.

[85] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1180–1189. JMLR.org, 2015.

[86] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

[87] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[88] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style, 2015.

[89] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

[90] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. In *Proceedings of the 28th British Machine Vision Conference (BMVC)*, 2017.

[91] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review, 2020.

[92] John Glover, Victor Lazzarini, and J. Timoney. Metamorph: Real-time high-level sound transformations based on a sinusoids plus noise plus transients model. In *Proceedings of the 15th Int. Conference on Digital Audio Effects (DAFx)*, 2012.

[93] Varun Gohil, S. Deepak Narayanan, and Atishay Jain. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *ReScience C*, 6(2), 2020. Accepted at NeurIPS 2019 Reproducibility Challenge.

[94] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[95] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014.

[96] Prachi Govalkar, Johannes Fischer, Frank Zalkow, and Christian Dittmar. A Comparison of Recent Neural Vocoders for Speech Signal Reconstruction. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 7–12, 2019.

[97] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, March 2012.

[98] John M. Grey. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(5):1270–1277, 1977.

[99] John M. Grey and John W. Gordon. Perceptual effects of spectral modifications on musical timbres. *The Journal of the Acoustical Society of America*, 63(5):1493–1500, 1978.

[100] D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[101] Eric Grinstein, Ngoc Q K Duong, Alexey Ozerov, and Patrick Pérez. Audio style transfer. In *ICASSP 2018 - 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, France, April 2018. IEEE.

[102] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. *AAAI Conference on Artificial Intelligence*, 2018.

[103] N. Hadad, L. Wolf, and M. Shahar. A two-step disentanglement method. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 772–780, 2018.

[104] Lamtharn Hantrakul, Jesse Engel, Adam Roberts, Chenjie Gu, and Lamtharn Hantrakul. Fast and flexible neural audio synthesis. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 524–530, Delft, The Netherlands, November 2019. ISMIR.

[105] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse H. Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. In Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos, editors, *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 50–57, 2018.

[106] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*, 2019.

[107] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

[108] Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. A functional taxonomy of music generation systems. *ACM Comput. Surv.*, 50(5), September 2017.

[109] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017.

[110] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc.

[111] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[112] A. Hines, J. Skoglund, A. Kokaram, and N. Harte. Visqol: The virtual speech quality objective listener. In *IWAENC 2012; International Workshop on Acoustic Signal Enhancement*, pages 1–4, 2012.

[113] Geoffrey E. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504 – 507, 2006.

[114] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[115] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[116] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey, 2020.

[117] Y. Hu and P. C. Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):229–238, 2008.

[118] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. In *International Conference on Learning Representations*, 2019.

[119] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3296–3297. IEEE Computer Society, 2017.

[120] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, Sageev Oore, and Roger B. Grosse. Timbretron: A wavenet(cycleGAN(CQT(audio))) pipeline for musical timbre transfer. In *International Conference on Learning Representations*, 2019.

[121] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, 2017.

[122] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[123] Eric J. Humphrey, Juan Pablo Bello, and Yann Lecun. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proc. ISMIR*, 2012.

[124] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[125] Julius O. Smith III. Viewpoints on the history of digital synthesis. In *Proceedings of the 1991 International Computer Music Conference, ICMC 1991, Montreal, Quebec, Canada, October 16-20, 1991*. Michigan Publishing, 1991.

[126] Julius O. Smith III and Xavier Serra. Overlap-add synthesis. `https://ccrma.stanford.edu/~jos/parshl/Overlap_Add_Synthesis.html`.

[127] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[128] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and J. Hershey. Single-channel multi-speaker separation using deep clustering. In *INTERSPEECH*, 2016.

[129] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[130] David A. Jaffe. Ten criteria for evaluating synthesis techniques. *Computer Music Journal*, 19(1):76–87, 1995.

[131] Zeyu Jin, Adam Finkelstein, Gautham J. Mysore, and Jingwan Lu. FFTNet: a real-time speaker-dependent neural vocoder. In *The 43rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018.

[132] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3365–3385, 2020.

[133] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[134] Nicolas Jonason, Bob L. T. Sturm, and Carl Thomé. The control-synthesis approach for making expressive and controllable neural music synthesizers. In *Proceedings of the 1st Joint Conference on AI Music Creativity*, page 9, Stockholm, Sweden, October 2020. AIMC.

[135] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2410–2419, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[136] Hanna Kamyshanska and Roland Memisevic. On autoencoder scoring. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 720–728, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[137] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv*, 2020.

[138] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *2018 IEEE Conference on Computer Vision and Pattern Recognition,*

CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 3907–3916. IEEE Computer Society, 2018.

[139] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms. In *Proc. Interspeech 2019*, pages 2350–2354, 2019.

[140] Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. Wavenode: A continuous normalizing flow for speech synthesis. In *ICML workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2020.

[141] J. W. Kim, R. Bittner, A. Kumar, and J. P. Bello. Neural music synthesis for flexible timbre control. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 176–180, 2019.

[142] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 161–165. IEEE, 2018.

[143] Sungwon Kim, Sang-Gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. FloWaveNet : A generative flow for raw audio. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3370–3378. PMLR, 09–15 Jun 2019.

[144] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1×1 convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 10236–10245, Red Hook, NY, USA, 2018. Curran Associates Inc.

[145] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[146] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[147] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 4743–4751. Curran Associates, Inc., 2016.

[148] Alexis Kirke and Eduardo Reck Miranda. A survey of computer systems for expressive music performance. *ACM Comput. Surv.*, 42(1), December 2009.

[149] Miranda E.R. Kirke A. An overview of computer systems for expressive music performance. *Guide to Computing for Expressive Music Performance. Springer, London.*, 2013.

[150] I. Kobyzev, S. Prince, and M. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[151] Ron Kohavi and Roger Longbotham. *Online Controlled Experiments and A/B Testing*, pages 922–929. 01 2017.

[152] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[153] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1863–1871, Bejing, China, 22–24 Jun 2014. PMLR.

[154] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 14910–14921. Curran Associates, Inc., 2019.

[155] Saïd Ladjal, Alasdair Newson, and Chi-Hieu Pham. A pca-like autoencoder, 2019.

[156] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic DENOYER, et al. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5963–5972, 2017.

[157] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR.

[158] E.L. Lehmann and J.P. Romano. *Testing statistical hypotheses*. Springer, 2005.

[159] Qi Lei, Ajil Jalal, Inderjit S. Dhillon, and Alexandros G. Dimakis. Inverting deep generative models, one layer at a time. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13910–13919, 2019.

[160] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Transactions on Multimedia*, 21(2):522–535, 2019.

[161] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabas Poczos. Mmd gan: Towards deeper understanding of moment matching network. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 2203–2213. Curran Associates, Inc., 2017.

[162] Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1718–1727. JMLR.org, 2015.

[163] Jen-Yu Liu, Yu-Hua Chen, Yin-Cheng Yeh, and Yi-Hsuan Yang. Unconditional Audio Generation with Generative Adversarial Networks and Cycle Regularization. In *Proc. Interspeech 2020*, pages 1997–2001, 2020.

[164] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 700–708, Red Hook, NY, USA, 2017. Curran Associates Inc.

[165] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 469–477. Curran Associates, Inc., 2016.

[166] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[167] Chien-Yu Lu, Min-Xin Xue, Chia-Che Chang, Che-Rung Lee, and Li Su. Play as you like: Timbre-enhanced multi-modal music style transfer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1061–1068, Jul. 2019.

[168] J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi. Understanding posterior collapse in generative latent variable models. In *International Conference on Learning Representations (ICLR)*, 2019.

[169] Sean Luke. *Computational Music Synthesis*. zeroth edition, 2019. Available for free at http://cs.gmu.edu/∼sean/book/synthesis/.

[170] Y. Luo and N. Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, 2019.

[171] Yin-Jyun Luo, Kat Agres, and Dorien Herremans. Learning Disentangled Representations of Timbre and Pitch for Musical Instrument Sounds Using Gaussian Mixture Variational Autoencoders. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 746–753, Delft, The Netherlands, November 2019. ISMIR.

[172] Y.J. Luo, K.W. Cheuk, T. Nakano, M. Goto, and D. Herremans. Unsupervised disentanglement of pitch and timbre for isolated musical instrument sounds. In *Proceedings of the International Society of Music Information Retrieval (ISMIR)*, 10/2020 2020.

[173] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

[174] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

[175] Pranay Manocha, Adam Finkelstein, Richard Zhang, Nicholas J. Bryan, Gautham J. Mysore, and Zeyu Jin. A differentiable perceptual audio metric learned from just noticeable differences. In *Interspeech*, October 2020.

[176] Rachel Manzelli, Vijay Thakkar, Ali Siahkamari, and Brian Kulis. Conditioning deep generative raw audio models for structured automatic music. In Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos, editors, *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 182–189, 2018.

[177] Andrés Marafioti, Nathanaël Perraudin, Nicki Holighaus, and Piotr Majdak. Adversarial generation of time-frequency features with application in audio synthesis. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4352–4362. PMLR, 09–15 Jun 2019.

[178] P. Martin. Comparison of pitch detection by cepstrum and spectral comb analysis. In *ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pages 180–183, 1982.

[179] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada. Deep griffin–lim iteration. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 61–65, 2019.

[180] Stephen McAdams. Timbre as a structuring force in music. *Proceedings of Meetings on Acoustics*, 19(1):035050, 2013.

[181] Stephen McAdams, Bruno Giordano, Patrick Susini, Geoffroy Peeters, and Vincent Rioux. A meta-analysis of acoustic correlates of timbre dimensions. *The Journal of the Acoustical Society of America*, 120(5):3275–3276, 2006.

[182] Stephen McAdams, Suzanne Winsberg, Sophie Donnadieu, Geert De Soete, and Jochen Krimphoff. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological research*, 58(3):177–192, 1995.

[183] Neil M. McLachlan. Timbre, pitch, and music. *OXFORD HANDBOOKS Linguistics, Language and Cognition, Psycholinguistics*, 2016.

[184] Muhammad Huzaifah Md Shahrin and Lonce Wyse. Applying visual domain style transfer and texture synthesis techniques to audio - insights and challenges. *Neural Computing and Applications*, 32, 02 2020.

[185] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. In *International Conference on Learning Representations*, 2017.

[186] Michael Michelashvili and Lior Wolf. Hierarchical timbre-painting and articulation generation. 2020.

[187] Parag K. Mital. Time domain neural audio style transfer. *NIPS workshop for Machine Learning for Creativity and Design*, 2017.

[188] Meinard Mller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer Publishing Company, Incorporated, 1st edition, 2015.

[189] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[190] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman. A universal music translation network. In *International Conference on Learning Representations*, 2019.

[191] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International*

Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 7176–7185. PMLR, 13–18 Jul 2020.

[192] Rui Nian, Jinfeng Liu, and Biao Huang. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers Chemical Engineering*, 139:106886, 2020.

[193] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

[194] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2642–2651, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[195] K. Oyamada, H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and H. Ando. Generative adversarial network-based approach to signal reconstruction from magnitude spectrogram. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2514–2518, 2018.

[196] M. Pal, R. Roy, J. Basu, and M. S. Bepari. Blind source separation: A review and analysis. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–5, 2013.

[197] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent. Filterbank design for end-to-end speech separation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6364–6368, 2020.

[198] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54 – 71, 2019.

[199] Marco Pasini. Melgan-vc: Voice conversion and audio style transfer on arbitrarily long samples using spectrograms, 2019.

[200] Kailash Patil, Daniel Pressnitzer, Shihab Shamma, and Mounya Elhilali. Music in our ears: The biological bases of musical timbre perception. *PLOS Computational Biology*, 8(11):1–16, 11 2012.

[201] J. Pätynen, V. Pulkki, and T. Lokki. Anechoic recording system for symphony orchestra. *Acta Acustica united with Acustica*, 94(6):856–865, 2008.

[202] F. Pedersoli, G. Tzanetakis, and K. M. Yi. Improving music transcription by pre-stacking a u-net. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 506–510, 2020.

[203] Geoffroy Peeters, Bruno L. Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams. The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, 2011.

[204] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI Conference on Artificial Intelligence*, New Orleans, United States, February 2018.

[205] Nathanaël Perraudin, Peter Balazs, and Peter L Søndergaard. A fast griffin-lim algorithm. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, pages 1–4. IEEE, 2013.

[206] Karol J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 1015–1018, New York, NY, USA, 2015. Association for Computing Machinery.

[207] Gustav Grund Pihlgren, Fredrik Sandin, and Marcus Liwicki. Improving image autoencoder embeddings with perceptual loss. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–7. IEEE, 2020.

[208] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. In *International Conference on Learning Representations*, 2019.

[209] Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. WaveFlow: A compact flow-based model for raw audio. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7706–7716. PMLR, 13–18 Jul 2020.

[210] R. Prenger, R. Valle, and B. Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621, 2019.

[211] Miller Puckette. *The Theory and Technique of Electronic Music*. World Scientific Publishing Co., Inc., USA, 2007.

[212] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019.

[213] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[214] Z. Rafii, A. Liutkus, F. R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo. An overview of lead and accompaniment separation in music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(8):1307–1335, Aug 2018.

[215] A. Ramires, P. Chandna, X. Favory, E. Gómez, and X. Serra. Neural percussive synthesis parameterised by high-level timbral features. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 786–790, 2020.

[216] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press Professional, Inc., USA, 1990.

[217] M. Ravanelli and Y. Bengio. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, 2018.

[218] R. Reed. Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747, 1993.

[219] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Comput. Surv.*, 2021.

[220] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

[221] Eitan Richardson and Yair Weiss. On gans and gmms. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 5847–5858. Curran Associates, Inc., 2018.

[222] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 833–840, Madison, WI, USA, 2011. Omnipress.

[223] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 2, pages 749–752 vol.2, 2001.

[224] C. Roads. Automated granular synthesis of sound. *Computer Music Journal*, 2:61, 1978.

[225] C. Roads. Introduction to granular synthesis. *Computer Music Journal*, 12(2):11–13, 1988.

[226] Fanny Roche, Thomas Hueber, Samuel Limier, and Laurent Girin. Autoencoders for music sound modeling : a comparison of linear, shallow, deep, recurrent and variational models. In University of Malaga (UMA), editor, *SMC 2019 - 16th Sound & Music Computing Conference*, number 1-6 in Proc. of SMC 2019, Malaga, Spain, May 2019.

[227] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 2234–2242. Curran Associates, Inc., 2016.

[228] C. N. D. Santos, Y. Mroueh, I. Padhi, and P. Dognin. Learning implicit generative models by matching perceptual features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4460–4469, 2019.

[229] Andy M. Sarroff and Michael A. Casey. Musical audio synthesis using autoencoding neural nets. In *Music Technology meets Philosophy - From Digital Echos to Virtual Ethos: Joint Proceedings of the 40th International Computer Music Conference, ICMC 2014, and the 11th Sound and Music Computing Conference, SMC 2014, Athens, Greece, September 14-20, 2014*. Michigan Publishing, 2014.

[230] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Commun. ACM*, 63(12):54–63, November 2020.

[231] D. Schwarz. Current research in concatenative sound synthesis. In *ICMC*, 2005.

[232] Diemo Schwarz. Corpus-based concatenative synthesis. *IEEE Signal Process. Mag.*, 24(2):92–104, 2007.

[233] Diemo Schwarz, Grégory Beller, Bruno Verbrugghe, and Sam Britton. Real-Time Corpus-Based Concatenative Synthesis with CataRT. In *9th International Conference on Digital Audio Effects (DAFx)*, pages 279–282, Montreal, Canada, September 2006. cote interne IRCAM: Schwarz06c.

[234] Diemo Schwarz, Grégory Beller, Bruno Verbrugghe, and Sam Britton. Real-time corpus-based concatenative synthesis with catart. In *International Conference on Digital Audio Effects (DAFx)*, pages 279–282, 2006.

[235] Allan Seago, Simon Holland, and Paul Mulholland. A critical analysis of synthesizer user interfaces for timbre. In Andy Dearden and Leon Watt, editors, *Proceedings of the XVIII British HCI Group Annual Conference HCI 2004*, volume 2, pages 105–108. Research Press International, Bristol, UK, 2004.

[236] Xavier Serra. *Musical Sound Modeling with Sinusoids plus Noise*, pages 91–122. Studies on New Music Research. Swets & Zeitlinger, 1997.

[237] Xavier Serra. *Musical Sound Modeling with Sinusoids plus Noise*, pages 91–122. Studies on New Music Research. Swets & Zeitlinger, 1997.

[238] Xavier Serra and J. Smith. Spectral modeling synthesis: A sound analysis/synthesis based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14:12–24, 1990. SMS.

[239] Kalpana Seshadrinathan, Thrasyvoulos N. Pappas, Robert J. Safranek, Junqing Chen, Zhou Wang, Hamid R. Sheikh, and Alan C. Bovik. Chapter 21 - image quality assessment. In Al Bovik, editor, *The Essential Guide to Image Processing*, pages 553 – 595. Academic Press, Boston, 2009.

[240] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783, 2018.

[241] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020.

[242] Kai Siedenburg, Ichiro Fujinaga, and S. McAdams. A comparison of approaches to timbre descriptors in music information retrieval and music psychology. *Journal of New Music Research*, 45:27 – 41, 2016.

[243] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[244] J. Smith and Xavier Serra. Parshl an analysis/synthesis program for non- harmonic sounds based on a sinusoidal representation. In *International Computer Music Conference*, pages 290–297, Urbana, Illinois, USA, 23/08/1987 1987.

[245] Julius O Smith. Virtual acoustic musical instruments: Review and update. *Journal of New Music Research*, 33(3):283–304, 2004.

[246] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3745–3753, Red Hook, NY, USA, 2016. Curran Associates Inc.

[247] Chunfeng Song, Feng Liu, Yongzhen Huang, Liang Wang, and Tieniu Tan. Auto-encoder based data clustering. In *Proceedings, Part I, of the 18th Iberoamerican Congress on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - Volume 8258*, CIARP 2013, page 117–124, Berlin, Heidelberg, 2013. Springer-Verlag.

[248] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[249] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 11918–11930. Curran Associates, Inc., 2019.

[250] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 574–584, Tel Aviv, Israel, 22–25 Jul 2020. PMLR.

[251] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. In Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos, editors, *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 334–340, 2018.

[252] Robert C. Streijl, Stefan Winkler, and David S. Hands. Mean opinion score (mos) revisited: Methods and applications, limitations and alternatives. *Multimedia Syst.*, 22(2):213–227, March 2016.

[253] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics.

[254] K. Subramani, P. Rao, and A. D'Hooge. Vapar synth - a variational parametric model for audio synthesis. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 796–800, 2020.

[255] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 3104–3112. Curran Associates, Inc., 2014.

[256] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[257] Hao Hao Tan, Y.J. Luo, and D. Herremans. Generative modelling for controllable audio synthesis of expressive piano performance. In *ICML Workshop on Machine Learning for Music Discovery (ML4MD)*, 2020.

[258] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. State of the Art on Neural Rendering. *Computer Graphics Forum (EG STAR 2020)*, 2020.

[259] Hoang Thanh-Tung and Truyen Tran. Toward a generalization metric for deep generative models. In *Advances in Neural Information Processing Systems*, 2020.

[260] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016. arXiv:1511.01844.

[261] T. Tolonen, V. Välimäki, and M. Karjalainen. Evaluation of modern sound synthesis methods. In *Scientific Report, ISBN 951-22-4012-2, ISSN 1239-1867*, 1998.

[262] I. Tolstikhin, S. Gelly, O. Bousquet, C. J. Simon-Gabriel, and B. Schölkopf. Adagan: Boosting generative models. In *Advances in Neural Information Processing Systems 30*, pages 5424–5433. Curran Associates, Inc., December 2017.

[263] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

[264] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. In *Third workshop on Bayesian Deep Learning (NeurIPS 2018)*, 2018.

[265] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4105–4113, 2017.

[266] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *J. Mach. Learn. Res.*, 17(1):7184–7220, January 2016.

[267] Harri Valpola. Chapter 8 - from neural pca to deep unsupervised learning. In Ella Bingham, Samuel Kaski, Jorma Laaksonen, and Jouko Lampinen, editors, *Advances in Independent Component Analysis and Learning Machines*, pages 143 – 171. Academic Press, 2015.

[268] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, page 125. ISCA, 2016.

[269] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel WaveNet: Fast high-fidelity speech synthesis. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3918–3926, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[270] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach,

R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6306–6315. Curran Associates, Inc., 2017.

[271] Sean Vasquez and Mike Lewis. Melnet: A generative model for audio in the frequency domain, 2019.

[272] Len Vande Veire, T. D. Bie, and J. Dambre. A cyclegan for style transfer between drum and bass subgenres. In *ICML workshop on Machine Learning for Media Discovery*, 2019.

[273] Gino Angelo Velasco, Nicki Holighaus, Monika Doerfler, and Thomas Grill. Constructing an invertible constant-q transform with nonstationary gabor frames. 09 2011.

[274] Prateek Verma and Julius O. Smith III. Neural style transfer for audio spectograms. In *NIPS workshop for Machine Learning for Creativity and Design*, 2017.

[275] T. Verma and Teresa H. Y. Meng. An analysis/synthesis tool for transient signals that allows a flexible sines+transients+noise model for audio. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, 6:3573–3576 vol.6, 1998.

[276] Tony S. Verma and Teresa H. Y. Meng. Extending spectral modeling synthesis with transient modeling synthesis. volume 24, page 47–59, Cambridge, MA, USA, July 2000. MIT Press.

[277] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.

[278] Bryan Wang and Yi-Hsuan Yang. Performancenet: Score-to-audio music generation with multi-band convolutional residual network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1174–1181, Jul. 2019.

[279] W. Wang, D. Yang, F. Chen, Y. Pang, S. Huang, and Y. Ge. Clustering with orthogonal autoencoder. *IEEE Access*, 7:62421–62432, 2019.

[280] X. Wang, S. Takaki, and J. Yamagishi. Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:402–415, 2020.

[281] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), June 2020.

[282] Yuxuan Wang, R. Skerry-Ryan, Daisy Stanton, Y. Wu, Ron J. Weiss, Navdeep Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, R. Clark, and R. A. Saurous. Tacotron: Towards end-to-end speech synthesis. In *INTERSPEECH*, 2017.

[283] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *arXiv preprint arXiv:1906.01529*, 2019.

[284] R. Wei, C. Garcia, A. El-Sayed, V. Peterson, and A. Mahmood. Variations in variational autoencoders - a comparative evaluation. *IEEE Access*, 8:153651–153670, 2020.

[285] Stefan Westerfeld. Spectmorph: Morphing the timbre of musical instruments. In *In Proceedings of the 16th International Linux Audio Conference (LAC)*, page 5—12, 2018.

[286] G. Widmer and W. Goebl. Computational models of expressive music performance: The state of the art. *Journal of New Music Research*, 33:203 – 216, 2004.

[287] Julia Wilkins, Prem Seetharaman, Alison Wahl, and Bryan Pardo. Vocalset: A singing voice dataset. In Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos, editors, *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 468–474, 2018.

[288] Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5), July 2020.

[289] Hanwei Wu and Markus Flierl. Vector quantization-based regularization for autoencoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6380–6387, Apr. 2020.

[290] Gus G. Xia and Shuqi Dai. Music style transfer: A position paper. In *Proceedings of the 6th International Workshop on Musical Metacreation*, page 6, Salamanca, Spain, June 2018. MUME.

[291] R. Yamamoto, E. Song, and J. Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203, 2020.

[292] Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. Multi-band melgan: Faster waveform generation for high-quality text-to-speech, 2020.

[293] Jinhyeok Yang, Junmo Lee, Young-Ik Kim, Hoon-Young Cho, and Injung Kim. Vocgan: A high-fidelity real-time vocoder with a hierarchically-nested adversarial network. In Helen Meng, Bo Xu, and Thomas Fang Zheng, editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 200–204. ISCA, 2020.

[294] Li-Chia Yang and Alexander Lerch. On the evaluation of generative models in music. *Neural Computing and Applications*, 32, 05 2020.

[295] B. Yegnanarayana and H. A. Murthy. Significance of group delay functions in spectrum estimation. *IEEE Transactions on Signal Processing*, 40(9):2281–2289, 1992.

[296] Serena Yeung, A. Kannan, Yann Dauphin, and Li Fei-Fei. Tackling over-pruning in variational autoencoders. *ArXiv*, abs/1706.03643, 2017.

[297] Bohan Zhai, Tianren Gao, Flora Xue, Daniel Rothchild, Bichen Wu, Joseph E. Gonzalez, and Kurt Keutzer. Squeezewave: Extremely lightweight vocoders for on-device speech synthesis, 2020.

[298] Dejiao Zhang, Yifan Sun, Brian Eriksson, and Laura Balzano. Deep unsupervised clustering using mixture of autoencoders. *CoRR*, abs/1712.07788, 2017.

[299] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[300] Y. Zhang and Z. Duan. Retrieving sounds by vocal imitation recognition. In *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2015.

[301] Zhaoyu Zhang, Mengyan Li, and Jun Yu. On the convergence and mode collapse of gan. In *SIGGRAPH Asia 2018 Technical Briefs*, SA '18, New York, NY, USA, 2018. Association for Computing Machinery.

[302] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models, 2017.

[303] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5885–5892, Jul. 2019.

[304] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[305] Michael Zhu and Suyog Gupta. To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018.

[306] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021.

[307] E. Zwicker. Subdivision of the audible frequency range into critical bands (frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248, 1961.