Neural Granular Sound Synthesis

First Author
Affiliation1
author1@myorg.org

Second Author
Affiliation2
author2@myorg.org

Third Author
Affiliation3
author3@myorg.org

ABSTRACT

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 space is bound by that of the descriptors, and its traversal is not continuously invertible to signal domain. Usually, such paths are randomized or hand-drawn, precluding structured temporality besides resynthesis.

In this paper, we demonstrate that generative neural networks can implement granular synthesis while alleviating some of its shortcomings. We efficiently replace the audio descriptor space 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 the grain library is no longer stored when generating. To learn structured paths inside this latent space, we add a higher-level temporal embedding trained on arranged sequences of grains. The model allows for analysis/synthesis with interpretable processing components and user controls.

This method can be applied to many types of libraries, including pitched or unpitched sounds such as musical notes and drums, as well as textures and environmental noises. We experiment with different creative tasks such as free-synthesis, one-shot sample generation and resynthesis.

1. INTRODUCTION

The process of generating musical audio has expanded into many fields with the support of computers. Some methods relying on parametric models, that may be derived from physical considerations on the sound source (eg. Karplus-Strong or wave guides), Fourier analysis (eg. sinusoids plus stochastic [1]) or signal processing operations (eg. frequency modulation). Alternatively to transformations on Fourier components, samplers and wavetables have been used to process input waveforms and generate audio. However, given large audio sample libraries, scaled methods are required to aggregate a model over the whole data and globally manipulate these features in the sound generation process. To this extent, corpus-based sound synthesis has been introduced by slicing a library of signals in shorter audio segments. According to some selection algorithms, these can be rearranged into new waveforms.

An instance of corpus-based methods is referred as granular sound synthesis, which uses waveform windows of short and fixed length. These units are called grains, and their size is usually set between 10 to 100 milliseconds. For a given library, all grains are analyzed through a set of audio descriptors [2] in order to facilitate their manipulation. Such higher-level dimensions structure a representation that reflects some form of local similarities across grains. A cloud of points whose distances relate to their underlying acoustic relationships. They may be matched to some target descriptor contours extracted from an other waveform, thus performing resynthesis in a manner akin to concatenative sound synthesis [3]. To a certain extent, its result can emulate the spectro-temporal dynamics of the considered signal. But the perceptual quality of such audio similarities, as assessed through a predefined set of acoustic descriptors, is inherently biased by their design. They only offer a limited consistency across many different sounds, within the corpus and with respect to other targets.

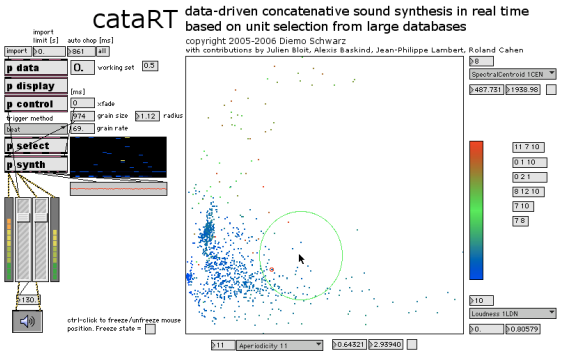

Figure 1. The explorative synthesis interface of CataRT. Grain space visualization with *Aperiodicity*, *Loudness* and *Spectral Centroid* features. (Figure from [3]).

Grain sequences may as well be drawn in more flexible ways, by sampling or free-synthesis trajectories in the acoustic descriptor space. Traversals across each of its dimensions map to grain series that are ordered according to the corresponding feature. However this space is not invertible, which means that it does not correspond to audio besides that of each of the scattered grains. The denser is the grain space, the smoother can be the corresponding assembled waveform. And as it only accounts for local relationships, it cannot generate the structured temporal dynamics of musical notes or drum hits without such driving target signal. For a given corpus, the choice of the analysis dimensions and the slicing size are essential parameters of

the grain space. They model the perceptual relationships across elements and set a trade-off: shorter grains may 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 to render a given signal duration. Moreover, as the granular synthesizer directly generates from the corpus snippets, its memory use is strictly proportional to the amount of data that is loaded and may quickly exceed ten of Gigabytes when considering nowadays sound sample library sizes. In a real-time setting, this causes further limitations to consider in a traditional granular synthesis space.

Drawing parallels between granular sound synthesis and latent variable models, we find that this method can be refined and efficiently applied to generative neural networks. Through the repeated observation of the grains, our proposed technique adaptively learns analysis dimensions which are continuously invertible to signal domain. Such space embeds the training dataset, which is no longer required in memory for generation. It as well 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. Our model addresses the aforementioned limitations, while relieving 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. These are continuous free-synthesis with variable step size, one-shot sample generation with controllable attributes and analysis/resynthesis for audio style transfer.

2. STATE OF THE ART

2.1 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 such a complex distribution that it 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 [4]. The goal of VI is to minimize differences between the approximated and real distribution, by using their Kullback-Leibler (KL) divergence:

$$q_\phi^*(\mathbf{z}|\mathbf{x}) = \underset{q_\phi(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}}{\operatorname{argmin}} \mathcal{D}_{KL}\big[q_\phi\left(\mathbf{z}|\mathbf{x}\right) \parallel p\left(\mathbf{z}|\mathbf{x}\right)\big] \quad (1)$$

By developing this divergence and re-arranging terms (detailed development can be found in [4]), we obtain:

$$\begin{aligned}\log p(\mathbf{x}) - D_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})\big] \\ = \mathbb{E}_{\mathbf{z}}\big[\log p(\mathbf{x}|\mathbf{z})\big] - \mathcal{D}_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big] \quad (2)\end{aligned}$$

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 (through the KL term). 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:

$$\mathcal{L}_{\theta,\phi} = \underbrace{-\mathbb{E}_{q_\phi(\mathbf{z})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big]}_{\text{reconstruction}} + \beta * \underbrace{\mathcal{D}_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big]}_{\text{regularization}} \quad (3)$$

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 $p_\theta(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Accordingly, a forward pass of the VAE consists in *encoding* a given data point $q_\phi : \mathbf{x} \rightarrow \{\mu(\mathbf{x}), \sigma(\mathbf{x})\}$ to obtain a mean $\mu(\mathbf{x})$ and variance $\sigma(\mathbf{x})$. These will allow us to obtain the latent $\mathbf{z}$ by sampling from the Gaussian, such that $\mathbf{z} \sim \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x}))$.

The representation learned with a VAE has a smooth topology [5] 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 [6] that it could be successfully applied to audio generation. Thus, it is the core of our neural model for granular synthesis of raw waveforms.

2.2 Neural waveform generation

Applications of generative neural networks to the raw audio waveform must face the challenge of modeling time series with very high sampling rates. In order to generate sound satisfyingly, the models must account for local features ensuring the audio quality, as well as aggregate longer-term relationships (consistent over ten of thousands of samples) in order to form meaningful signals. Based on the causal nature of audio, auto-regressive models have first been employed. These models decompose the joint distribution over the whole waveform $\mathbf{x} = \{x_1 \dots x_T\}$ into a product of conditional distributions. Hence, each sample is generated conditioned on all previous ones:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t|x_1 \dots x_{t-1}) \quad (4)$$

Amongst these models, WaveNet [7] has been established as the reference solution for high-quality speech synthesis. This model has also been successfully applied to musical domains such as Nsynth [8]. 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. Some of these limitations can be alleviated through a longer and overall more complicated training scheme [9].

More specific to musical audio generation, the Symbol-to-Instrument Neural Generator (SING) proposed an overlap-add convolutional architecture [10] on top of which is trained a sequential embedding S over frame steps $\mathbf{F}_{1\dots f}$ and instrument, pitch, velocity classes $(\mathbf{I}, \mathbf{P}, \mathbf{V})$. Considering a window size of 1024 with a 75% overlap, the number of recurrent iterations $1 \dots f$ is thus roughly divided by 256 before the forward pass of the up-sampling convolutional decoder D. Hence, given an input signal of log-spectrogram magnitude $l(\mathbf{x}) = \log(\epsilon + |\text{STFT}[\mathbf{x}]|^2)$, the decoder reconstruction $\hat{\mathbf{x}}$ minimizes the objective:

$$\underset{D,S}{\text{argmin}} ||l(\mathbf{x}), l(\hat{\mathbf{x}})||_1 \tag{5}$$

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

Recently, additional postulates arising from digital signal processing knowledge have specified tighter constraints on model definitions, leading to high sound quality with lowered training costs. In this spirit, the Neural Source-Filter (NSF) model [11] applies the idea of Spectral Modeling Synthesis (SMS) [1] to speech synthesis. Its input module receives acoustic features and computes conditioning information for both source and temporal filtering modules that jointly output a waveform. In order to render both voiced and unvoiced sounds, a sinusoidal and gaussian noise excitations are fed into separate filter modules. Through stacked convolutions, these signals are processed and mixed after filtering. Estimation of noisy and harmonic components is further improved by relying on a multi-scale spectral reconstruction criterion over a set of Short-Term Fourier Transform (STFT) magnitudes.

Similar to NSF, but for pitched musical audio, the Differentiable Digital Signal Processing (DDSP [12]) library 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 frequency domain reverberation module that refines the acoustic quality of the signal and increases the modularity of the model. Although this process offers very promising results, it is restricted in the nature of signals that can be generated.

3. NEURAL GRANULAR SOUND SYNTHESIS

The previously proposed models have developed a strong audio prior knowledge for neural synthesis that we as well pursue. In this paper, we propose a model that can learn both a frame-level representation and modeling at multiple time scales, as depicted in Figure 3. The audio quality of short-term synthesis features 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 the latent granular space, by explicitly reconstructing waveforms through an *overlap-add mechanism* across audio grain sequences. The synthesis operations are suited to any type of spectrogram reconstruction, while being interpretable. Furthermore, our proposal allows for analysis prior to data-driven resynthesis, as well as continuous and variable length free-synthesis trajectories. Taking advantage of its frame-level representation, a higher-level sequence embedding can be trained to generate audio events with meaningful temporal structure. In its less restrictive definition, our model allows for unconditional sampling. However, it can be trained with additional independent controls (such as pitch or other user classes) to support more explicit interactions for composition and sound transfer.

3.1 Grain latent space

Formally, we consider a set $\mathcal{X}$ of audio grains $\mathbf{x} \in \mathbb{R}^{d_x}$ extracted from a given sound corpus, with fixed grain size d_x. This dataset follows an underlying probability distribution $p(\mathbf{x})$ that we aim to approximate through a continuous mapping p_θ. We could then consistently synthesize novel examples with $p_\theta \sim p(\mathbf{x})$. As for granular synthesis, this likelihood is hardly evaluated in the waveform domain, but enhanced through a condensed set of descriptors $\mathbf{z} \in \mathbb{R}^{d_z}$ $(d_z \ll d_x)$. This low-dimensional space relates the most salient features of the data, such as acoustic descriptors, or the latent variables of a generative neural network. Such representation can be learned using an encoder network to compute $\mathbf{z} = q_\phi(\mathbf{x})$ paired with a decoder so that $p_\theta(q_\phi(\mathbf{x})) \sim \mathbf{x}$ as estimated in every $\mathbf{x} \in \mathcal{X}$. However, we have no guarantee that these latent variables continuously mirror the target distribution $p(\mathbf{x})$. In order to learn such smooth latent distribution $p(\mathbf{z})$, we use a Variational Auto-Encoder [4] to perform probabilistic latent inference and sampling with a Gaussian prior.

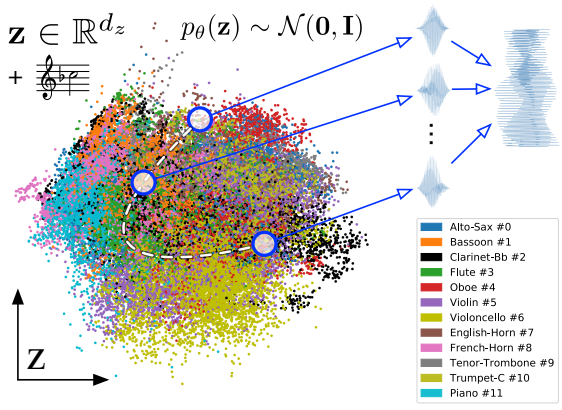

Figure 2. 2D visualization of a learned grain space over individual pitched notes. Using latent dimensions of $p_\theta(\mathbf{z})$ as a substitute for acoustic descriptors allows for an invertible representation. Grains can be synthesized from any position, such as along a continuous free-synthesis path, and overlap-add into a waveform. A pitch target can be added.

3.2 Analysis and synthesis with neural networks

To train on overlap-add reconstruction, the model processes series of g successive grains $\mathbf{s_x} = \{\mathbf{x}_1 \dots \mathbf{x}_g\}$ sliced from a given waveform $\mathbf{w}_{1..T}$. The ratio between its duration T and g is given by the hop size separating neighboring grains. Each of these grains $\mathbf{x}_i$ is analyzed by the encoder as $q_\phi : \mathbf{x}_i \rightarrow \{\mu(\mathbf{x}_i), \sigma(\mathbf{x}_i)\}$ to form a corresponding series $\mathbf{s_z} = \{\mathbf{z}_1 \dots \mathbf{z}_g\}$ of latent coordinates:

$$\mathbf{z}_i = \mu(\mathbf{x}_i) + \eta * \sigma(\mathbf{x}_i) \tag{6}$$

with $\eta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The input layers of the encoder are strided residual convolutions that successively down-sample the input grains through their temporal 1-dimensional filters. Their output is then fed into several fully-connected linear layers that map to Gaussian means and variances at the desired latent dimensionality $d_z \ll d_x$.

Given a latent series $\mathbf{s_z}$, the decoder must synthesize every corresponding grains prior to overlap-add. To this extent, each $\mathbf{z}_i$ is mapped with residual fully-connected layers to 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. This technique adapts the filter design proposed in [12] to granular synthesis. To this extent, we replace the recurrence of the DDSP decoder over envelope features by separate forwards over overlapping grain features. Denoting the Discrete Fourier Transform DFT and its inverse iDFT, it roughly amounts to computing $\hat{\mathbf{X}}_i = \mathbf{H}_i * \text{DFT}(\mathbf{n}_i)$ and $\hat{\mathbf{x}}_i = \text{iDFT}(\hat{\mathbf{X}}_i)$ (details can be found in [12]). Since the DFT of a real valued signal is Hermitian, symmetry implies that for an even grain size d_x, the network only needs to output filter coefficients for the $d_h = d_x/2 + 1$ positive frequencies.

These grains are arranged with overlap-add into a waveform $\hat{\mathbf{w}}$ which is passed through a final post-processing module (inspired from [13]). It applies a multi-channel convolution (unit stride) that learns a parallel set of time-invariant FIR filters and improves the audio quality of the assembled signal.

3.3 Multi-resolution training objective

The size d_h of the filtering module is directly related to the grain size, while the temporal resolution is set by the hop size. These parameters can be interpreted as the spectrogram accuracy that the model can achieve. To optimize the waveform reconstruction, we employ a multi-resolution spectral loss [11, 12]. 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 [14] on which we compare log-magnitudes $l(\mathbf{w}) = \log(\epsilon + |\text{STFT}[\mathbf{w}]|^2)$ with the absolute distance $||.||_1$. In addition to fit multiple settings $\text{STFT}_{1\dots N}$, we can control the trade-off between low and high-energy components (eg. harmonics) with the ϵ floor value [10]. Since our model structures a granular space, KL regularization and sampling (6) are individually applied to each latent point $\mathbf{z}_i$. According to our model, the original VAE objective (3) is adapted as:

$$\mathcal{L}_{\theta,\phi} = \underbrace{\sum_{n=1}^{N} ||l_n(\mathbf{w}), l_n(\hat{\mathbf{w}})||_1}_{\text{reconstructions}} + \beta * \underbrace{\sum_{i=1}^{g} \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}_i|\mathbf{x}_i) \parallel p_\theta(\mathbf{z})\right]}_{\text{regularizations}} \tag{7}$$

3.4 Sequence embedding

In order to sample audio events spanning series of grains with a consistent temporal structure, a higher-level embedding $s_\theta(\mathbf{e} \in \mathbb{R}^{\mathbf{d_e}})$ can be added to the granular latent space. It efficiently uses this intermediate frame-level representation in order to learn the longer-term relationships. This is achieved with a temporal recurrent neural network trained on ordered sequences of grain features $\mathbf{s_z}$. It can be applied to musical notes or drums with attack, decay, sustain and release. As a result, our proposal can as well synthesize meaningful paths inside its granular latent space. Such process is as well probabilistic, by first sampling from the Gaussian $\mathbf{e} \sim \mathcal{N}(0, \mathbf{I})$, then sequentially decoding:

$$s_\theta : \mathbf{e} \rightarrow \hat{\mathbf{s}}_\mathbf{z} \tag{8}$$

and finally generating grains and overlap-add waveform with $p_\theta : \hat{\mathbf{s}}_\mathbf{z} \rightarrow \hat{\mathbf{s}}_\mathbf{x} \rightarrow \hat{\mathbf{w}}$.

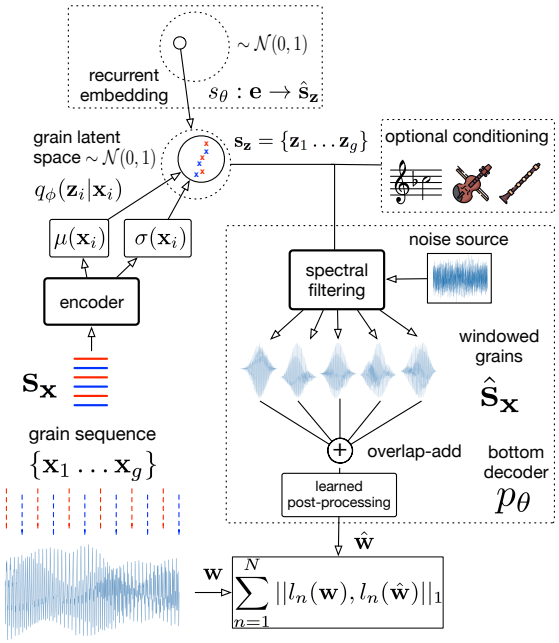

Figure 3. Overview of our proposed neural granular model.

4. EXPERIMENTS

4.1 Datasets

In order to generate across a variety of sound domains, we train models on the following datasets:

1. *Studio-On-Line* sampled at 22050 Hz, provides individual note recordings 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. [15]

2. *Violin and Cello solo* sampled at 22050 Hz, provides a free-improvisation recording of each instrument for a total of around 2 hours.

3. *8 Drums* sampled at 16000 Hz, around 6000 one-shot samples in *Clap, Cowbell, Crash, Hat, Kick, Ride, Snare, Tom* instrument classes. [1]

4. *Methlab samples* sampled at 16000 Hz, a pack of 260 *Drum and Bass* samples in *Bass, Break, Cymbal, Fx, Hat, Kick, Lead, Pad, Perc, Snare* classes.

5. *10 animals*: sampled at 22050 Hz, around 3 minutes of recordings for each of *Cat, Chirping Birds, Cow, Crow, Dog, Frog, Hen, Pig, Rooster, Sheep* classes of the ESC-50 dataset for Environmental Sound Classification. [2]

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 $[128, 256, 512, 1024, 2048]$. 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 $\epsilon = 5e^{-3}$ (Figure 4). The grain latent space has dimensionality $d_z = 96$ and the recurrent embedding uses $d_e = 256$.

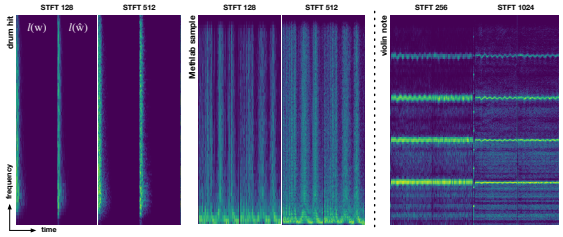

Figure 4. Log-magnitude and linear-frequency spectrograms computed for different STFT window sizes. Left are a drum hit and a *Methlab* clip sampled at 16000 Hz and right is a violin note at 22050 Hz. Input and the reconstruction are denoted by $l(\mathbf{w})$ and $l(\hat{\mathbf{w}})$.

4.2 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. If considering 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 rendered sound attributes:

$$p_\theta : (\hat{\mathbf{s}}_\mathbf{z}, \mathbf{oh}_{\text{class}}) \rightarrow \hat{\mathbf{s}}_\mathbf{x}^{\text{cond.}} \rightarrow \hat{\mathbf{w}}^{\text{cond.}} \tag{9}$$

4.3 Training

The model is trained according to (7). 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 8 hours on a single GPU, such as an Nvidia Titan V.

5. RESULTS

To assess the generative qualities of the model, we provide audio samples of data reconstructions as well as examples of neural granular sound synthesis [3] . They refine the common tasks of free-synthesis through continuous latent interpolations, controllable data-driven resynthesis and sampling structured audio events with target attributes. These generations are performed either unconditionally or with different controls according to datasets and possible composition aims.

5.1 Audio reconstruction and one-shot samples

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 a higher-level sequence embedding according to (8). For use in composition (eg. MIDI score), this sampling can be done with conditioning over user classes such as pitch and target instrument (9).

5.2 Interpolations in the grain latent space

The latent path of a grain series with temporal structure (eg. a musical note) may have any shape. And the acoustic features of each encoded dimension are usually not disentangled, in other words not interpretable. However since the VAE learns a continuously invertible grain space, it can be explored with smooth interpolations that render free-synthesis trajectories. Some multidimensional latent curves that are mapped to overlap-add grain sequences (Figure 2). Given two random samples from the latent Gaussian prior $\mathbf{z}_1, \mathbf{z}_g \sim \mathcal{N}(0, \mathbf{I})$, we perform a linear interpolation as:

$$\mathbf{s}_\mathbf{z}^{\text{linear}} = \mathbf{z}_1 * (1 - \alpha) + \mathbf{z}_g * \alpha \tag{10}$$

with g uniform steps $\alpha = [\![0 \dots 1]\!]$. Given two orthogonal unit vectors $\mathbf{u_1} \perp \mathbf{v_1} \in \mathbb{R}^{d_z}$ and a radius $r > 0$, we draw a circular interpolation as:

$$\mathbf{s}_\mathbf{z}^{\text{circular}} = r * \cos\alpha * \mathbf{u_1} + r * \sin\alpha * \mathbf{v_1} \tag{11}$$

with angular steps $\alpha = [\![0 \dots 2\pi]\!]$. If varying the radius $r = [\![r_1 \dots r_g]\!]$ (with $r_1 \neq r_g$), we obtain a spiral interpolation. When repeating forward and backward traversals of a linear interpolation or looping a circular curve, we can modulate non-uniformly the steps α between latent points in order to bring additional expressivity to the synthesis. Free-synthesis can be performed at variable lengths

[1] https://github.com/chrisdonahue/wavegan/tree/v1

[2] https://github.com/karolpiczak/ESC-50

[3] https://anonymized124.github.io/neural_granular_synthesis/

(in multiples of g) by concatenating several contiguous latent paths.

5.3 Audio style transfer

To perform data-driven resynthesis, a target sample is analyzed by the encoder. Its corresponding latent features are then decoded. Since the input content may differ from the training data distribution, latent features only account partially for the target sample. Moreover, the decoder may only generate audio qualities that are consistent with its training dataset. Thus emulating the target sound to a certain extent and in the style of the learned grain space. Using conditioning over *timbre* (eg. instrument classes) allows for finer control over such audio style transfer. 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.

5.4 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* [4] server controlled from a *MaxMsp* [5] patch. For instance a neural drum machine [3] featuring a step-sequencer driving a model with sequential embedding and conditioning trained over the *8 Drums* dataset classes.

6. 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 adresses 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.

Acknowledgments

anonymized

7. REFERENCES

[1] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, p. 12–24, 1990.

[2] G. Peeters and al., "The Timbre Toolbox: Extracting audio descriptors from musical signals," *The Journal of the Acoustical Society of America*, vol. 130, no. 5, pp. 2902–2916, 2011.

[3] D. Schwarz and al., "Real-Time Corpus-Based Concatenative Synthesis with CataRT," in *Proceedings of the International Conference on Digital Audio Effects*, 2006.

[4] D. P. Kingma and al., "Auto-encoding variational bayes," *International Conference on Learning Representations*, 2014.

[5] I. Higgins and al., "beta-VAE: Learning basic visual concepts with a constrained variational framework," *International Conference on Learning Representations*, 2016.

[6] P. Esling and al., "Generative timbre spaces with variational audio synthesis," in *Proceedings of the International Conference on Digital Audio Effects*, 2018.

[7] A. van den Oord and al., "Wavenet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[8] J. Engel and al., "Neural audio synthesis of musical notes with WaveNet autoencoders," *International Conference on Machine Learning*, 2017.

[9] A. van den Oord and al., "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," *International Conference on Machine Learning*, 2018.

[10] A. Defossez and al., "Sing: Symbol-to-instrument neural generator," *Advances in Neural Information Processing Systems*, vol. 31, p. 9041–9051, 2018.

[11] X. Wang and al., "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2019.

[12] J. Engel and al., "DDSP: Differentiable Digital Signal Processing," *International Conference on Learning Representations*, 2020.

[13] C. Donahue and al., "Adversarial audio synthesis," *International Conference on Learning Representations*, 2019.

[14] K. W. Cheuk and al., "nnAUDIO: a pytorch audio processing tool using 1d convolution neural networks," *International Society for Music Information Retrieval*, 2019.

[15] G. Ballet and al., "Studio online 3.0: An internet "killer application" for remote access to ircam sounds and processing tools," *Journee Informatique Musicale*, 1999.

[4] https://pypi.org/project/python-osc/

[5] https://cycling74.com