VAPAR SYNTH - A VARIATIONAL PARAMETRIC MODEL FOR AUDIO SYNTHESIS

Krishna Subramani, Preeti Rao(Guide)

Electrical Engineering, IIT Bombay
Roll Number : 150020008

ABSTRACT

With the advent of data-driven statistical modeling and abundant computing power, researchers are turning increasingly to deep learning for audio synthesis. These methods try to model audio signals directly in the time or frequency domain. In the interest of more flexible control over the generated sound, it could be more useful to work with a parametric representation of the signal which corresponds more directly to the musical attributes such as pitch, dynamics and timbre. We present VaPar Synth - a Variational Parametric Synthesizer which utilizes a conditional variational autoencoder(CVAE) trained on a suitable parametric representation. We demonstrate our proposed model's capabilities via the reconstruction and generation of instrumental tones with flexible control over their pitch.

1. INTRODUCTION

Early work in audio synthesis relied on instrument and signal modeling approaches (physical and spectral modeling synthesis). Recently, there has been interesting work in the use of generative models, broadly labelled 'Neural Audio Synthesis'. These methods rely on the ability of algorithms to extract musically relevant information from vast amounts of data. Various approaches such as autoregressive modeling, Generative Adversarial Networks and VAEs have been proposed with varying degrees of success given the ultimate goal of modeling complex instrument sound sources.

Sarroff et al. [1] were among the first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra. They were inspired to use model an autoencoder using neural networks(NNs) because, given 'enough data', these networks could learn mappings from higher dimensional spaces to lower dimensional spaces, which could be perceptually relevant to audio synthesis. Their main motivation to use the spectral(FFT) based representation was that it could be inverted back into the time domain(assuming the phase information is preserved). Their investigations revealed a 'graininess' in the reconstructed sound. They also claim that the use of a deep model by simply stacking more layers in the architecture does not necessarily improve the quality of the reconstructed audio.

Roche et al. [2] extended this analysis. With the release of NSynth [3], they were able to harness the large number of instrument recordings to experiment different autoencoder architectures, namely variational and recurrent autoencoders. They experimented with the network parameters for optimal reconstruction, and also analyzed the so called 'latent space' which is essentially a low dimensional representation of the input data. They also analyze the usability of this latent space in the interpolation of sounds.

One limitation acknowledged by the above authors was the lack of meaningful control over the latent space for use in synthesis. Esling et al. [4] incorporated a regularization term in the VAE latent space in order to effect some control over the perceptual timbre of synthesized instruments. Through their experiments, they were able to 'generate' audio by sampling points from this latent space, and could also perform interpolation in this space.

A common drawback of the previous methods is the lack of phase during the reconstruction process, leading to an inherently lossy reconstruction of the audio. Another issue is the frame-wise analysis-synthesis based reconstruction procedure, which does not take into account the temporal evolution of the signal. Inspired by WaveNets [5] success in modelling speech autoregressively, Engel et al. [3] were inspired to extend this analysis to musical instruments as well. To account for the longer temporal range of music, they replaced the basic spectral autoencoder by a WaveNet autoencoder. This allows them to morph meaningfully between instruments, thus creating sounds that are realistic and expressive. However, their model is very complex, and requires a large dataset to train. Also, because of the nature of sampling, it is time consuming to generate audio(1 second of audio at 16 kHz takes a few minutes to generate!)

Wyse [6] also had the similar goal of modeling instrument audio in an autoregressive fashion. However he wanted to achieve audio synthesis of musical instruments with the target characteristics provided as an external input. Thus, he trained a Recurrent Neural Network to predict waveform samples more accurately by providing additional information like pitch, velocity and instrument class. The experiments were designed to check the models ability to synthesize good quality audio and to generalize to parameters the model has not been trained on. The inability of the model to generalize to notes with pitches the network has not seen before was a limitation.

In the present work, rather than generating new timbres,

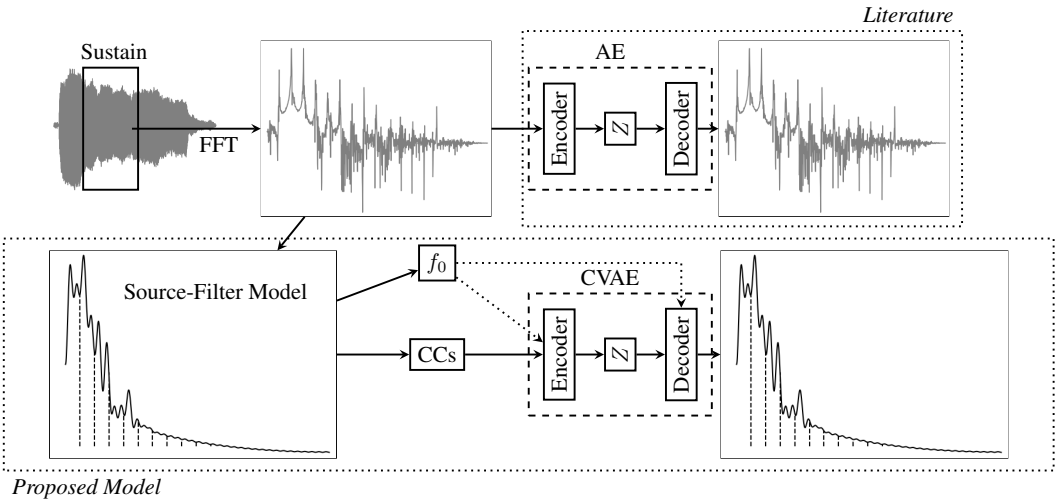


Fig. 1: Flowchart of the state of the art frame-wise audio synthesis pipeline (upper branch) and our proposed model (lower branch). Z represents the latent space learned by the (CV)AE.

we consider the problem of synthesis of a given instrument's sound with flexible control over the pitch and loudness dynamics. As is well known, pitch shifting without timbre modification (i.e. preserving naturalness of the instrument sound from body resonances) requires the use of a source-filter decomposition where the filter (i.e. the spectral envelope) is kept constant during pitch transposition [7]. The other advantages of such a powerful parametric representation over raw waveform or spectrogram is the potential to achieve high quality with less training data. Recognizing this in the context of speech synthesis, Blaauw et al. [8] used a vocoder representation for speech, and then trained a VAE to model the frame-wise spectral envelope. To test the generative capabilities of the model, they sample from the latent space and analyze the reconstructed spectrum. They also experiment with interpolation in the latent space to better understand it, and they conclude that the use of the parametric representation shows promise in generation.

The VAE has the attractive properties of continuous latent variables and the additional control over the latent space by way of prior probabilities giving good reconstruction performance [9]. Our approach in this paper is to use the VAE for the modeling of the frame-wise spectral envelope similar to Blaauw et al. [8] but for instrumental sounds. Given that even for a chosen instrument, the spectral envelope is not necessarily invariant with changing pitch, we further explore the conditional VAE (CVAE), to achieve conditioning of the generation on pitch. The motivation for our work comes from the desire to synthesize realistic sounds of an instrument at pitches that may not be available in the training data. Such a context can arise in styles such as Indian art music where continuous pitch movements are integral parts of the melody. We evaluate our approach on a dataset of violin, a popular instrument in Indian music, adopted from the West, due to its human voice-like timbre and ability to produce continuous pitch movements [10].

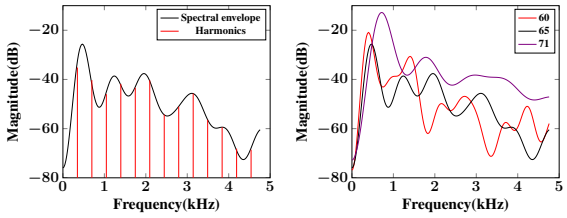
The parametric representation we adopt involves source filter decomposition applied to the harmonic component of the spectrum extracted by the harmonic model [11]. The filter is estimated as the envelope of the harmonic spectrum and represented via low-dimensional cepstral coefficients [12]. Therefore, as opposed to training a network to directly reconstruct the full magnitude spectrum as currently done in previous literature (upper branch in Figure 1), we train a CVAE on the real cepstral coefficients (CCs) conditioned on the pitch (lower branch in Figure 1). The trained network will presumably capture the implicit relationship between source and filter from the dataset samples and thus generalize better to new conditional parameter settings.

2. DATASET

We work with the good-sounds dataset [13]. It consists of two kinds of recordings(individual notes and scales) for 12 different instruments sampled at $F_s = 48kHz$. We select the violin subset of the data. The recordings have been labeled for played as good(hence good-sounds!) and bad. We use only 'good' recordings, all of which are played in mezzo-forte loudness on a single violin. We choose to work with the 4^{th} octave(MIDI 60-71) representing mid-pitch range. Figure 2 shows the instances(recordings) per note in the selected octave. The average duration is about $4.5s$ per note. From each note segment, we first extract the sustained portion by applying energy thresholds. We split the data to train(80%) and test(20%) instances across MIDI note labels. We train our model with frames(duration $21.3ms$) from the train instances, and evaluate system performance with frames from the test instances.

MIDI	60	61	62	63	64	65
Instances	28	26	26	26	26	26
MIDI	66	67	68	69	70	71
Instances	34	30	30	26	26	26

Fig. 2: Instances per note in the overall dataset



(a) Single note (MIDI 65) (b) Different notes

Fig. 3: Spectral envelopes from the parametric model

3. PROPOSED SYSTEM

3.1. The Parametric Model

From the frame-wise magnitude spectrum, we obtain the harmonic representation using the harmonic plus residual model [11](currently, we neglect the residual). Next, we decompose the harmonic spectrum with the source-filter model as proposed by Caetano and Rodet [14]. The filter is represented by the "spectral envelope". Roebel et al. [7] outline a procedure to extract the harmonic envelope using the 'True Amplitude Envelope(TAE)' algorithm (originally by Imai [15]). This method addresses the issues with the traditional cepstral liftering, where the envelope obtained tends to follow the mean energy. The TAE iteratively applies cepstral liftering to push the envelope to follow the spectral peaks. The envelope is represented by the cepstral coefficients(CCs), with the number of kept coefficients(K_{cc}) dependent on the pitch (fundamental frequency f_0) and sampling frequency as below,

$$K_{cc} \leq \frac{F_s}{2f_0}. \tag{1}$$

Figure 3a shows a spectral envelope extracted from one frame of a MIDI 65 instance superposed on the actual harmonics. We see that the TAE provides a smooth function from which we can accurately estimate harmonic amplitudes by sampling at the harmonic frequency locations.

The spectral envelopes for different notes appear in Figure 3b indicating clear differences in the spectral envelope shape across the range of pitches shown. This reinforces the importance of taking into account spectral envelope dependence on the pitch for musical instruments [16, 14]. It is expected that the process of estimation of the envelope itself also could contribute to the variation. Incorporating the dependency on pitch will obtain more realistic harmonic amplitude estimates and potentially more natural synthesized sound over what is possible with a phase vocoder.

3.2. AE, VAE and CVAE

We try out two kinds of Networks - Autoencoders(AE) and Conditional Variational Autoencoders(CVAE) [9, 17, 18]. Before describing our model, we present a brief description of these models.

3.2.1. Autoencoder

The name 'Autoencoder' is self-explanatory. It consists of a network that will learn to automatically encode data into a more compact space which in turn brings about an efficient reconstruction of the original data. A rather extensive comparison of several AE network structures can be found in [2]. The general structure of such a network is represented Figure 4. The input is same as the output, and the network is trained to minimize the reconstruction loss(usually the mean squared error(MSE)) between the input and output,

$$\mathcal{L} = \left\| X - \hat{X} \right\|^2, \tag{2}$$

where $\hat{X}$ is the network's reconstruction of the input X. In our case, the network input is the cepstral coefficients, and thus MSE represents a perceptually relevant distance in terms of squared error between the input and reconstructed log magnitude spectral envelopes. Because of its bottlenecked shape, the AE is forced to learn a compact(lower dimensional) representation(or 'code') for the input data. What we would ideally like is for the AE to learn a latent space over which we can exercise some kind of control. However, the objective function enforces nothing like this, and very often, the latent space is sparse.

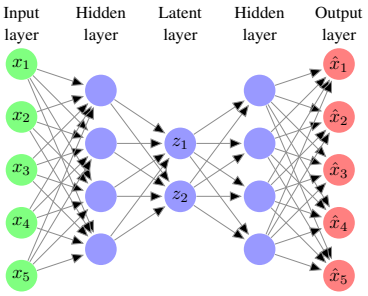


Fig. 4: AE Structure

3.2.2. Variational Autoencoder

The Variational Autoencoder (VAE) is loosely based on an AE. However, the working principle and optimization criteria are quite different. VAE's are inspired from Variational Inference, which have their roots in Probabilistic Graphical Models. Figure 5 shows a graphical model highlighting the dependence of X on z.

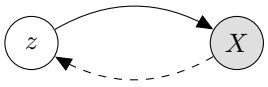


Fig. 5: Graphical Model showing dependence of X on z

From Figure 5, we can write the following,

$$P(X, z) = P(X|z)P(z),$$

However, we are interested in how z depends on X. For this, we have to compute,

$$P(z|X) = \frac{P(X|z)P(z)}{P(X)}.$$

To obtain $P(X)$, the following integral has to be computed:

$$P(X) = \int_z P(X|z, \theta)P(z)\mathrm{d}z,$$

which becomes intractable for high dimensional z. Thus, Variational inference aims to approximate $P(z|X)$ with another distribution $Q(z|X)$. To ensure that the approximation is good enough, you minimize the KL-Divergence between the two distributions,

$$D_{KL}[Q(z|X, \theta)||P(z|X, \theta)] = \mathbb{E}_{z\sim Q}\{\log Q(z|X, \theta) - \log P(z|X, \theta)\}, \quad (3)$$

where θ indicates that the distribution is parametrized by θ and $\mathbb{E}$ is the expectation operator under Q. We drop the θ afterwards to lighten the expressions. Because a sufficiently general function can map a Normal distribution to any desired distribution, the prior on the latent variable z is chosen to be Normally distributed. Equation 3 can be rewritten using Bayes rule, and the terms can be rearranged to obtain:

$$\log P(X) - D_{KL}[Q(z|X)||P(z|X)] = \mathbb{E}_{z\sim Q}\{\log P(X|z)\} - D_{KL}[Q(z|X)||P(z)]. \quad (4)$$

We want to maximize the left side of that equation as we want to maximize $\log P(X)$. If $Q(z|X)$ is a good approximation of $P(z|X)$, the KL-divergence will be small and we will actually maximize the log-likelihood.

To make all terms in Equation 4 tractable, $Q(z|X)$ is chosen to be a Normal distribution whose parameters are learned from the data by the encoder. The right-hand side can then be maximized using stochastic gradient descent. Sometimes, instead of using the optimization objective as mentioned in Equation 4, it is useful to weigh the terms relative to each other by multiplying the KL Divergence term with a weighting factor [2, 4],

$$\mathcal{L} = \mathbb{E}_{z\sim Q}\{\log P(X|z)\} - \beta D_{KL}\{Q(z|X)||P(z)\}. \quad (5)$$

If $\beta = 1$, it reduces to a normal VAE. Small β gives more importance to the MSE term, thus prioritizing perfect reconstruction(making the VAE behave more like an AE). Similarly, high β strongly enforces the prior to be Normal at the expense The choice of β is a hyperparameter that has to be decided. A more complete and mathematically rigorous analysis of VAEs can be found in [17, 9].

3.2.3. Conditional VAE

Data generation requires sampling from the trained VAE. The procedure is actually rather straightforward because of the Normality imposed on the latent space. Indeed, creating new outputs simply requires sampling from a Normal distribution and passing it through the decoder. However, VAE's does not allow much control over the sampling procedure. Unless we have access to a representation of the latent space, we are always going to use a sampled z that will produce an output "similar" to the training data. Say for example, our input data is multimodal, and we want to generate samples corresponding to a particular mode. This will not be possible until we know what part of the latent space generates which mode(ideally though, the VAE as presented should not be used for multimodal data, as the Normal prior is unimodal in itself!). CVAEs [18] address this issue by modifying Equation 4 to model the conditional distribution of the input on a conditioning variable. The motivation to use a CVAE over a AE in our context is two fold:

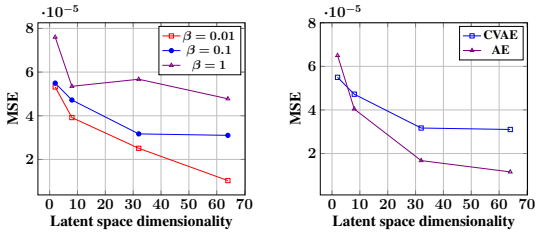
1. It allows us to obtain a continuous latent space from which we can sample points(and synthesize the corresponding audio).

2. By conditioning on the pitch, we expect the network to capture the subtle dependencies between the timbre and the pitch, thus allowing us to generate the envelope more accurately, and at the same time giving us the ability to control the pitch.

3.3. Network Architecture

The main hyperparameters in our networks are the dimensionality of the latent space and the value of β. To decide these, we train the network on the train data instances with different hyperparameters, and evaluate the networks MSE with the test instances. The MSE reported here is the average reconstruction error across all the test instances. Figure 6a shows the MSE for $\beta = [0.01, 0.1, 1]$. From our trials, we conclude $\beta = 0.1$ to be the sweet spot to tradeoff between both of these. With this value of β, Figure 6b shows the MSE plots for AE and CVAE for latent space dimensions $[2, 8, 32, 64]$. We note a steep fall until 32, becoming more flat later, indicating that 32 is a good choice for latent space dimensionality.

All networks are implemented in PyTorch [19]. For both AE and cVAE, we work with a similar network architecture - an



(a) CVAE, varying β (b) CVAE($\beta = 0.1$) vs AE

Fig. 6: MSE plots to decide hyperparameters

encoder having dimensions : $[91, 91, 32]$, and a decoder having the same(but reversed) architecture. According to Equation 1, different pitches can have different number of CCs. 91 is the number of CCs for the lowest pitch(MIDI 60). For the higher pitches, the CCs are zero padded to the same dimension 91. All the layers are linear fully connected layers and use leaky ReLu activations to allow for stable training. The optimization was performed using ADAM [20] with an initial learning rate of 10^{-3}, and training was run for 2000 epochs with a batch size of 512 on an NVIDIA GeForce GTX 1070 Mobile GPU.

4. EXPERIMENTS

While we present results in this paper for the AE and CVAE on the parametric representation of the frame-wise spectrum, we have also carried out the similar experiments directly with the frame magnitude spectrum. As expected the reconstruction performance is relatively poor; the complete results are reported in our accompanying notebook[1]. We perform the following 2 kinds of experiments to demonstrate the capabilities of our model.

1. Reconstruction - We omit all instances of certain selected pitches during training, and see how well our model can reconstruct a note of the unseen target pitch. The spectral envelope of a note instance of the target pitch is input to the network. The output of the network is the reconstructed envelope, to be evaluated with respect to the input.

2. Generation - The purely 'synthesis' aspect of our model; we see how well our model can generate note instances with new unseen pitches.

4.1. Reconstruction

We consider two distinct training contexts for the reconstruction of a note with unseen pitch. (a) all instances of the neighbouring MIDI notes upto 3 neighbours are included in the training set, as shown in Figure 7; this is performed for $T = [63, 64, 65, 66, 67, 68]$. (b) the training set contains instances of only the octave endpoint MIDI notes, 60 and 71; we reconstruct instances of all the intermediate notes.

MIDI	T - 3	T - 2	T - 1	T	T + 1	T + 2	T + 3
Kept	✓	✓	✓	×	✓	✓	✓

Fig. 7: ✓ indicate MIDI note instances included in the training set for the synthesis of a given target note of MIDI label T.

In each of the above cases, we compute the MSE in the frame-wise spectral envelope match across all frames of all the target instances. The results are presented in Figure 8. We can see that the CVAE produces better reconstruction, especially when the target pitch is far from the pitches available in the training data. In the latter case, the MSE is seen to decrease as the target pitch moves closer to its nearer octave end pitch in both networks, as one might expect. Overall, the conditioning provided by the CVAE helps to capture the pitch dependency of the spectral envelope more accurately.

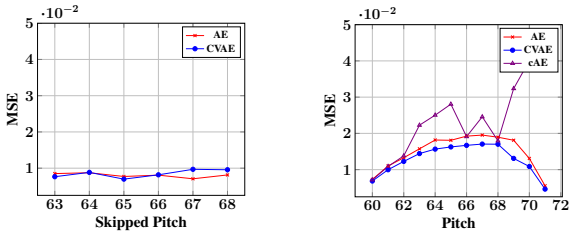


Fig. 8: Spectral envelope MSE across unseen pitch note instances with close MIDI neighbours in training data (left), and only octave end notes in training data (right).

To emulate the effect of pitch conditioning with an AE, we train the AE by appending the pitch to the input CCs and reconstructing this appended input as shown in Figure 9. We were motivated to do this from Wyse [6] who followed a similar approach of appending the conditional variables to the input of his model. For reconstruction, we do not work with the reconstructed f_0', rather we use the original f_0 given as an input. This way, the AE is comparable to our proposed CVAE in that the network might potentially learn something from the f_0 during reconstruction.

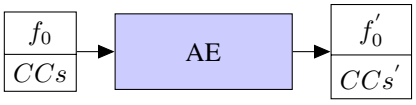


Fig. 9: 'Conditional' AE(cAE)

We then perform the experiment of training this model only on the endpoints(as mentioned above in subsection 4.1). The right plot in Figure 8 shows the MSE plot obtained for the cAE. As can be seen, appending f_0 does not seem to be improving the model, on the contrary, it seems to worsen the AEs performance in reconstructing the skipped notes.

[1]Notebook : `https://www.ee.iitb.ac.in/student/~krishnasubramani/icassp2020.html`

We also carry out the reconstruction experiments with the frame magnitude spectra as detailed by Roche et al. [2]. Figure 10 shows the input spectrogram to the model. It is a MIDI 63 note.

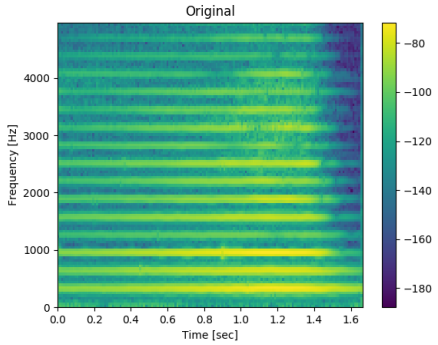


Fig. 10: Original Magnitude(dB) Spectrogram

We consider two cases - (a) As shown in Figure 7 by skipping T = 63 and training on the 3 adjacent notes, and (b) Including MIDI 63 with the neighbouring pitches. As expected from an AE, it can reconstruct the note it has been trained on reasonably well. Figure 11 shows the reconstructed spectrogram. The harmonic structure is preserved.

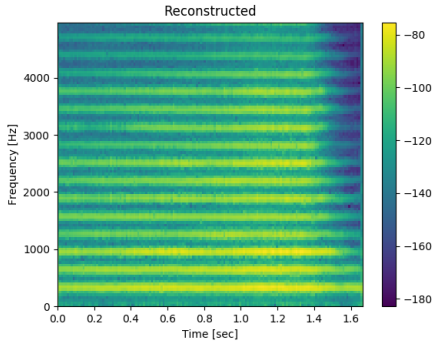


Fig. 11: Reconstructed Magnitude(dB) Spectrogram

However, the AE fails to reconstruct the note it has not been trained on, inspite of having been trained on nearby notes. Figure 12 shows the reconstructed spectrogram. It is quite distorted and lacks even a clear harmonic structure.

4.2. Generation

The previous experiment evaluated the networks reconstruction capabilities. However, we are ultimately interested in using it as a synthesizer. Thus, in this experiment, we see how well the network can generate the spectral envelope of an instance of a desired pitch (not available in the training data of the network). For this, we train on instances across the entire

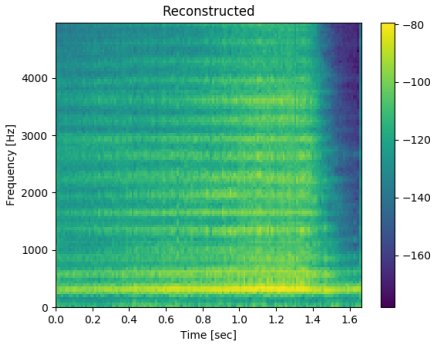


Fig. 12: Reconstructed Magnitude(dB) Spectrogram

octave sans MIDI 65, and then generate MIDI 65. Generation comes naturally to the CVAE, as we just have to sample points from the prior distribution, and pass them through the decoder along with the conditional parameter f_0 to generate the spectral envelope(lower branch in Figure 1). Since a single latent variable represents a single frame, we have to coherently sample multiple latent variables and decode them to obtain multiple contiguous frames. Our approach to sample points from the latent space is motivated from [8] i.e. we perform a random walk with a small step size near the origin in the latent space to sample points coherently. We synthesize the audio by sampling the envelope at the harmonics of the specified f_0, and perform a sinusoidal reconstruction. We do not have an objective measure to evaluate the quality of the generated note. However informal listening indicates that it sounds close to the natural violin sound except for the missing soft noisy sound of the bowing. Incorporating residual modeling in the parametric representation in future would help restore this.

Further, we try to generate a practically useful output, viz. a vibrato violin note with typical vibrato parameters. This exercise involves reconstructing spectral envelopes corresponding to the continuum in the neighbourhood of the note MIDI pitch. In this case as well, the generated vibrato tone sounded natural in informal listening. More formal subjective listening tests are planned involving also synthesis of larger pitch movements or melodic ornaments from Indian raga music. It must be recalled that we have not taken loudness dynamics into account. All our dataset instances were labeled mezzo-forte. However a more complete system will involve capturing spectral envelope dependencies on both pitch and loudness dynamics.

5. CONCLUSION

The goal of this work was to explore autoencoder frameworks in generative models for audio synthesis of instrumental tones. We critically reviewed recent approaches and identified the problem of natural synthesis with flexible pitch control. We then presented VaPar synth - our model to generate audio. Through our parametric representation, we can decouple the

'timbre' and 'pitch', and can thus rely on the network to model the inter-dependencies. We use a variational model as it gives us the ability to directly sample points from the latent space. Moreover, by conditioning on the pitch, we can generate the learnt spectral envelope for that pitch (something which would not be possible in a vanilla VAE), thus giving us the power to vary the pitch contour continuously in principle. We then present a few experiments demonstrating the capabilities of our model. To the best of our knowledge, we have not come across any work using a parametric model for musical tones in the neural synthesis framework, especially exploiting the conditioning function of the CVAE.

6. REFERENCES

[1] Andy M Sarroff and Michael A Casey, "Musical audio synthesis using autoencoding neural nets," in *ICMC*, 2014.

[2] Fanny Roche, Thomas Hueber, Samuel Limier, and Laurent Girin, "Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models," *arXiv preprint arXiv:1806.04096*, 2018.

[3] 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*. JMLR. org, 2017, pp. 1068–1077.

[4] Philippe Esling, Adrien Bitton, et al., "Generative timbre spaces: regularizing variational auto-encoders with perceptual metrics," *arXiv preprint arXiv:1805.08501*, 2018.

[5] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[6] Lonce Wyse, "Real-valued parametric conditioning of an rnn for interactive sound synthesis," *arXiv preprint arXiv:1805.10808*, 2018.

[7] Axel Roebel and Xavier Rodet, "Efficient Spectral Envelope Estimation and its application to pitch shifting and envelope preservation," in *International Conference on Digital Audio Effects*, Madrid, Spain, Sept. 2005, pp. 30–35, cote interne IRCAM: Roebel05b.

[8] Merlijn Blaauw and Jordi Bonada, "Modeling and transforming speech using variational autoencoders.," in *Interspeech*, 2016, pp. 1770–1774.

[9] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[10] Chris Haigh, *Indian violin*, 2014 [Accessed: 21-Oct-2019], `http://www.fiddlingaround.co.uk/india/`.

[11] Xavier Serra et al., "Musical sound modeling with sinusoids plus noise," *Musical signal processing*, pp. 91–122, 1997.

[12] Marcelo Caetano and Xavier Rodet, "Musical instrument sound morphing guided by perceptually motivated features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1666–1675, 2013.

[13] Oriol Romani Picas, Hector Parra Rodriguez, Dara Dabiri, Hiroshi Tokuda, Wataru Hariya, Koji Oishi, and Xavier Serra, "A real-time system for measuring sound goodness in instrumental sounds," in *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[14] Marcelo Caetano and Xavier Rodet, "A source-filter model for musical instrument sound transformation," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 137–140.

[15] S. IMAI, "Spectral envelope extraction by improved cepstrum," *IEICE*, vol. 62, pp. 217–228, 1979.

[16] Wayne Slawson, "The color of sound: a theoretical study in musical timbre," *Music Theory Spectrum*, vol. 3, pp. 132–141, 1981.

[17] Carl Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[18] Kihyuk Sohn, Honglak Lee, and Xinchen Yan, "Learning structured output representation using deep conditional generative models," in *Advances in neural information processing systems*, 2015, pp. 3483–3491.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.