

Wavetable Synthesis Using CVAE for Timbre Control Based on Semantic Label

Tsugumasa Yutani*, Yuya Yamamoto*, Shuyo Nakatani†, Hiroko Terasawa*

* University of Tsukuba, Tsukuba, Japan

E-mail: tsugumasa320@gmail.com, s2130507@u.tsukuba.ac.jp, terasawa@slis.tsukuba.ac.jp

† Cybozu Labs, Tokyo, Japan

E-mail: nakatani@labs.cybozu.co.jp

Abstract—Synthesizers are essential in modern music production. However, their complex timbre parameters, often filled with technical terms, require expertise. This research introduces a method of timbre control in wavetable synthesis that is intuitive and sensible and utilizes semantic labels. Using a conditional variational autoencoder (CVAE), users can select a wavetable and define the timbre with labels such as *bright*, *warm*, and *rich*. The CVAE model, featuring convolutional and upsampling layers, effectively captures the wavetable nuances, ensuring real-time performance owing to their processing in the time domain. Experiments demonstrate that this approach allows for real-time, effective control of the timbre of the wavetable using semantic inputs and aims for intuitive timbre control through data-based semantic control. Audio examples are available on the online demo page¹.

I. INTRODUCTION

Synthesizers, electronic musical instruments that create sounds using analog or digital signals, play a significant role in modern music production and performance. Although various methods of synthesizing sounds exist, effectively controlling the synthesis algorithm and obtaining the desired timbre requires specialized knowledge and experience. This complexity arises because numerous synthesis parameters are intricate, low-level, and unrelated to the perceptual and semantic properties of the timbre. This challenge is often called the synthesizer programming problem[1], which can even hinder the creative process for experienced musicians and professionals[2].

To address this problem, the concept of extracting synthesizer features has been proposed. [3]–[6]. This approach involves extracting features from a user-supplied sound (the target sound) and providing macro-controls that enable users to adjust the timbre based on these features, thereby simplifying the process of achieving the desired timbre.

Our research employs wavetable synthesis (WTS), one of the earliest software synthesis methods developed by Max Mathews in the late 1950s[7]. WTS, the core technology behind many synthesis methods, remains popular in numerous software synthesizers today. WTS works by cyclically referencing and replicating a waveform (WT: Wavetable) to produce a sound. The timbre is typically modified using filters, amplifiers, or morphing between different wavetables(WTs).

Furthermore, recent years have seen significant advancements in deep generative models. Among these, autoencoder

(AE) [8] and variational autoencoder (VAE) [9] have gained prominence in various fields. These models compress the input data into a compact, low-dimensional latent space and reconstruct the original data from this compressed representation, facilitating the extraction of valuable feature representations from the data. Additionally, conditional variational autoencoder(CVAE)[10], which enables conditional generation based on specified conditions, has also been introduced.

Our research uses a CVAE to generate a WT for a single cycle, which is then applied to the oscillator within WTS. We have engineered a model that allows users to define the desired timbre through semantic labels. Figure 1 depicts a schematic overview of this process. In previous studies, methods for generating WT using deep generative models have been proposed[3], [11], [12]. However, issues regarding the diversity, controllability, and accuracy of the generated WT remain to be solved. Diverging from these methods, our research proposes a technique that enables users to emphasize or attenuate specific frequency ranges within a given WT using semantic labels, similar to audio effects such as filters, equalizers, and enhancers using CVAE. This approach represents the primary contribution of our research.

II. RELATED WORKS

A. Synthesizer Programming Problem

The synthesizer programming problem, as outlined in Section I, is a complex issue that experienced musicians and professionals face. This challenge stems from the intricate control mechanisms and the necessity for specialized expertise, particularly in the realm of synthesizer sound creation [2]. Substantial research has focused on automating the estimation of synthesis parameters to address this issue. As a result, two approaches have emerged: the first aims to approximate synthesis parameters to replicate a target sound [13]–[16] and the second utilizes descriptive language concerning timbre for parameter estimation [17]–[19].

The feature synthesizer methodology warrants particular attention. This method is swift and user-friendly, specifically designed for music production and live performances. It involves extracting features from target timbres and providing macro-controls, thereby facilitating the adjustment of timbre based on these extracted features. This approach significantly simplifies the process of achieving the desired timbre. The

¹<https://noto.li/XZKK4q>

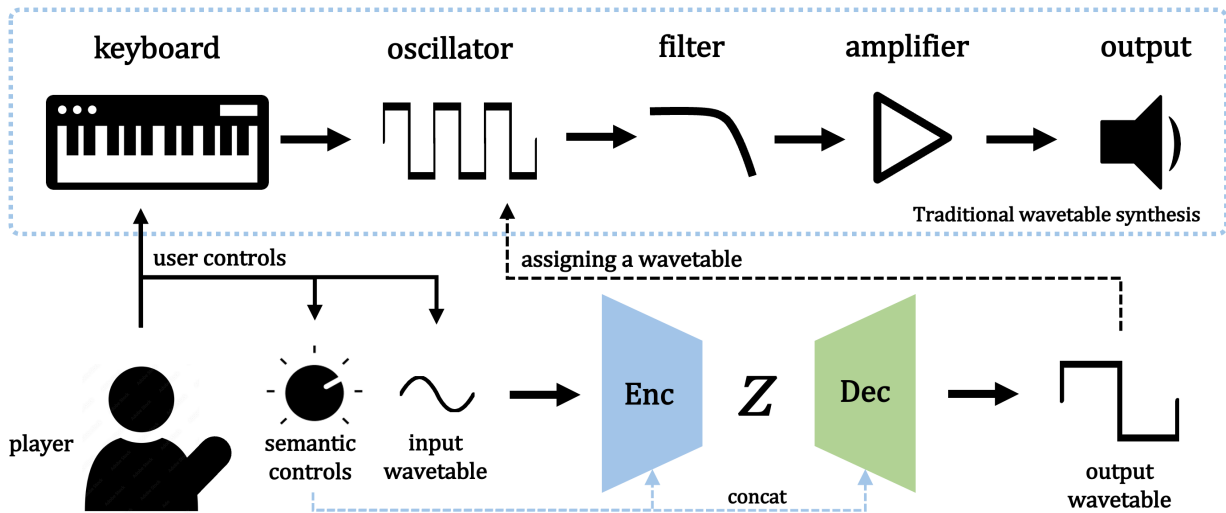


Fig. 1. Schematic showing an overview of the wavetable synthesis (WTS) method demonstrated in this research. A participant creates a wavetable (WT) exhibiting a specific timbre, achieved by conditioning the initial WT with various semantic labels. This generated WT is consistently referenced and emitted as a continuous sound. In this WTS setup, the pitch and duration can be adjusted using a keyboard or similar devices, akin to the conventional WTS operations. Additionally, the timbre of the WT can be modified by using filters and amplifiers.

efficacy of the feature synthesis method in enabling intuitive timbre modification has been substantiated in various research studies[3]–[5], [11], [20]. Our research focuses on this feature synthesis approach, emphasizing its ability to rapidly adapt to changes in timbre during music production and performance.

B. Neural Audio Synthesis

Neural audio synthesis is a technology used for generating or transforming audio data using deep learning. As outlined in Section I, deep generative models are rapidly developing and have become an essential technology in music. Research to generate audio data is also active, and research for applications to synthesizers has been proposed[21]–[23].

Several approaches have been proposed to complement the existing synthesis methods using deep learning. To solve the synthesizer programming problem, Esling et al. [24] showed that VAE can simultaneously process parameter estimation, timbre control, and reconstruction with semantic labels in a single model for subtractive synthesis. Kreković [3] illustrated the feasibility of processing and controlling parameter estimation, timbre control, and reconstruction with semantic labels in a unified WTS model, highlighting the generation of WT from semantic labels.

This research explores the application of deep learning to musical synthesis, akin to the earlier research. It focuses on the latest technical trends in model construction, loss functions, and methods for calculating semantic labels.

C. Advances in Wavetable Synthesis

Wavetable synthesis generates a sound by repeatedly referencing a waveform for one cycle. This method, widely used in various genres, faces the common challenge of timbre generation, as with other synthesis methods. Creating a WT generally involves editing the waveform and frequency spectrum [25] or choosing a preset WT.

However, while these methods significantly impact timbre, they often diverge from human intuition. For instance, subtle changes in the shape or frequency spectrum of a WT can alter its timbre. Nonetheless, discerning the characteristics of the timbre through visual inspection is challenging. Even when selecting a WT from a preset, users must remember the relationship between the name of the preset and its corresponding timbre, which requires a certain level of expertise. Consequently, achieving the desired timbre in WTS can be difficult.

To address this problem, as outlined in Section I, previous research generated WT using VAE[12] and AE[11], and the timbre was changed by randomly manipulating the latent space. Furthermore, Kreković [3] proposed a model that generates WT from three input labels: *bright*, *warm*, and *rich*, using a decoder [26]. However, embedding WT features into a limited number of latent spaces (labels) poses challenges. The diversity and accuracy of the generated WT are still two issues that remain to be solved. To enhance the diversity and accuracy of the WT that can be generated, we have proposed a model that uses CVAE[10]. This model is designed to create WT that exhibits a one-period waveform characteristic.

III. METHODS

A. CVAE

The CVAE[10], an extension of the VAE, is a generative model. This model is notably effective in generating data conditioned on a specific variable, denoted as c . In our research, we incorporated the condition, c , into the encoder and the decoder to apply these conditions effectively.

The input waveform, x , and condition, c , (in this research, WT and semantic labels, respectively) were mapped to the latent variable, z , by the encoder, $q_\phi(z|x, c)$. Subsequently, the decoder, $p_\theta(x|z, c)$, generated the output data, \hat{x} .

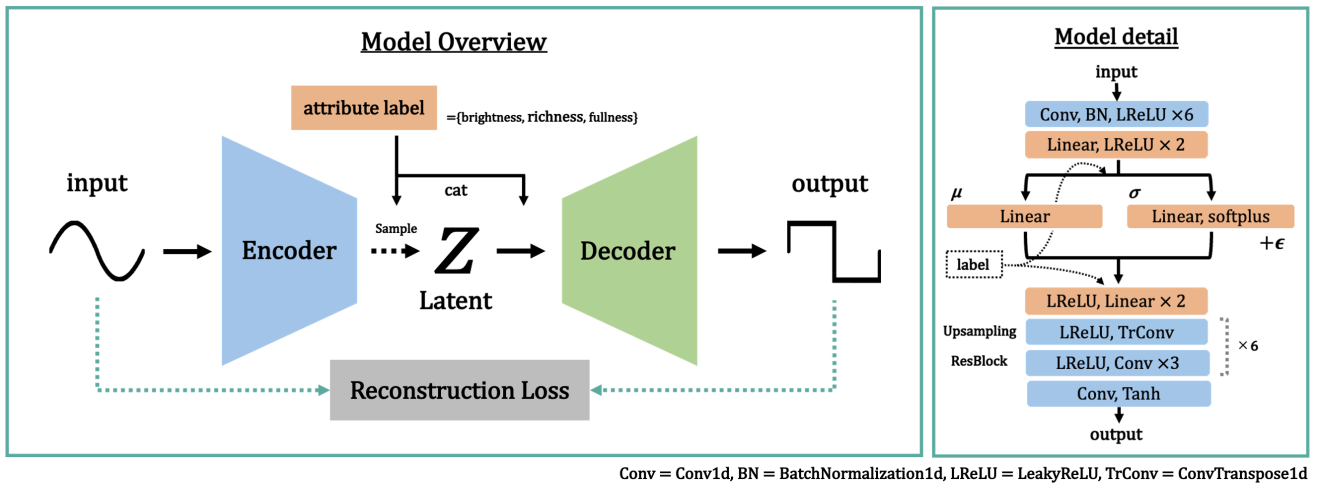


Fig. 2. Schematic of the model configuration. Left: Model overview, right: detailed configuration.

B. Loss Function

The loss function comprised the reconstruction loss and the Kullback–Leibler (KL) divergence term. These components were optimized by minimizing them through the stochastic gradient descent (SGD) method. The loss function in this research was defined as follows:

$$L_{CVAE}(x) = \mathbb{E}_{\hat{x} \sim p_{\theta}(x|z)}[d(x, \hat{x})] + \beta \times D_{KL}[q_{\phi}(z|x, c)||p(z)] \quad (1)$$

The first term in Equation (1), the reconstruction loss, was used for evaluating the accuracy of the generated data in reconstructing the original data and aimed to minimize the information loss between the encoder and decoder. Previous research has suggested using multiscale spectral distance to estimate the waveform distances[21], [23]. Due to the short data length in our WT approach, we calculated the spectral distance using only the maximum table size (3600 samples in our research) after iterative concatenation.

The function, d , for the reconstruction loss was defined as follows:

$$d(x, \hat{x}) = \frac{\|S_x - S_{\hat{x}}\|_F}{\|S_x\|_F} + \log \|S_x - S_{\hat{x}}\|_1 \quad (2)$$

where S_x and $S_{\hat{x}}$ are the amplitude spectra of x and \hat{x} , respectively. Here, $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|_1$ the L1 norm.

The amplitude spectrum of the input waveform, x , and the output waveform, \hat{x} , were compared and their loss was calculated. This approach is based on the assumption that the perceptual characteristics of the synthesized sound remain almost invariant between the signal waveforms that are out of phase. Following the convention adopted by Kreković [3], we improved the frequency resolution of the spectrum by concatenating six WTs iteratively.

The second term of Equation (1), the KL divergence term, measured the closeness of the latent variable, z , sampled from the encoder, $q_{\phi}(z|x, c)$, to the predefined multivariate normal distribution, $p(z)$. This term regularized the latent space, ensuring that the model captured the general data characteristics instead of merely memorizing the data.

The hyperparameter, β , adjusted the balance between the reconstruction loss and the KL divergence term. The setting of this parameter significantly influenced the performance of the model, as proposed by Higgins et al. [27].

C. Model Configuration

The architecture of the model was adapted from a structure outlined in a previous research[23] and was modified to enable the conditional generation of WT. The encoder combined the convolutional layers that convert the WT into a 32-dimensional latent representation. The decoder used an upsampling layer and a residual network[28] to generate the WT. Conditional generation was executed by concatenating the relevant labels to the inputs and outputs within the latent space[12]. Figure 2 illustrates an overview of the model structure.

D. Computation of Semantic Labels

Previous research has explored the computation of semantic labels in sounds, including WTS. In our research, we employed the labels *bright*, *warm*, and *rich*, following the methodology outlined by Kreković [3], [29]. These labels were derived from the spectral centroid, the energy ratio of the odd harmonics, and the spectral density, followed by normalization.

The spectral centroid, representing the center of mass of the spectrum, indicates the balance between the high and low frequencies and correlates with the timbre brightness[30]. The energy ratio of the odd harmonics, i.e., the proportion of energy in the odd harmonics compared to other harmonics, was associated with the warmth of the timbre[31]. The spectral density, which is calculated as the standard deviation of the spectrum, corresponded to the richness of the timbre[32].

For extracting features from a single WT, six WTs were concatenated in succession to improve the spectral resolution, following the method outlined in Section III-B. Audio features were computed using a fast Fourier transform and a Hanning window. Since the derived values do not naturally correspond with human perception, a transformation was applied to ensure linear alignment with human perception, particularly within the

range of [0, 1]. However, the energy ratio of the odd harmonics for the *warm* label, inherently falling within [0, 1], required no additional normalization.

IV. EXPERIMENTS

A. Dataset

The data length directly influences the capability of representing higher frequency components in WTS. More extensive data lengths enable a more comprehensive representation of these components. In prior research, Kreković employed datasets with 320 samples [11] and 327 samples [3], whereas Hyrkas et al. utilized a dataset of 512 samples [12].

In this research, we utilized a dataset comprising 600 samples, which exceeds the size of these previously employed datasets. The dataset was partitioned into three segments: 80 % for training, 10 % for validation, and 10 % for testing. This distribution ensured a comprehensive evaluation of the model across different data subsets.

B. Training

Training was conducted by feeding the model with training data and the corresponding labels. The Adam optimizer[33] was used with a learning rate of 1e-4. The batch size was 32, and the number of epochs was 30000 to minimize the reconstruction and KL divergence.

Over-optimizing the KL divergence between the prior and posterior distributions in the early stages of training can lead to an issue known as KL vanishing[34], in which the latent variables are ignored in the reconstruction. To mitigate this issue, the variable, β , for the KL divergence was gradually increased according to the following Equation:

$$\beta(e, w, \beta_{\min}, \beta_{\max}) = \begin{cases} \beta_{\max} & \text{if } e > w \\ \exp(t \cdot (\log \beta_{\max} - \log \beta_{\min}) + \log \beta_{\min}) & \text{otherwise} \end{cases} \quad (3)$$

$$t = \frac{e}{w} \quad (4)$$

where e is the current number of epochs, w is the upper bound on the number of epochs, β_{\min} is the minimum value set for β , β_{\max} is the maximum value set for β , starting from $\beta_{\min} > 0$ and increasing exponentially to β_{\max} exponentially as the learning progresses. In this research, $w = 10000$, $\beta_{\min} = 1e - 4$, $\beta_{\max} = 1e - 1$, and the period of increase was 1 epoch.

V. EVALUATION

The reconstruction quality of the generated WT and the controllability and generation time of the conditioning generation were evaluated. This research aimed to enhance the intuitiveness of timbre control and enable real-time timbre searches. To achieve this, the following three steps were essential: 1) high-quality reconstruction of the WT input to the model 2) conditional generation according to the values of the semantic labels, and 3) short processing time for generation and real-time use. The results obtained by performing the abovementioned steps are shown below. The source code of

the model and the weights of the trained model used to obtain the results are also available²

A. Reconstruction Quality

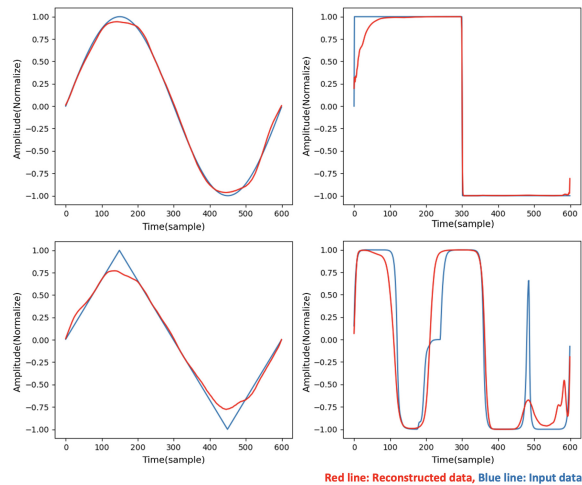


Fig. 3. Example of the qualitative evaluation of the reconstruction quality. The plots show that the reconstruction is done according to the input data.

The mean absolute error (MAE) was calculated to evaluate the reconstruction quality between the reconstructed waveform and the original input waveform. The calculated MAE amounted to 0.082 for the waveform state and 0.013 for the logarithmic amplitude spectrum. The logarithmic amplitude spectrum was calculated by repeatedly concatenating the six WTs as described in Section III-B to facilitate a comparison between the frequency components. The reconstructed waveforms are shown in Figure 3, which reproduce the input waveforms.

These results indicate that the reconstructed waveform successfully emulates the input waveform, capturing the essential characteristics of the WT. Future work will entail a comparison of our method with other models and conducting listening experiments to validate the reconstruction quality quantitatively.

B. Controllability of conditioning generation

To evaluate the controllability of the conditioning generation, the MAE and Pearson correlation coefficients of the output labels concerning the input labels (condition c) were checked. The output labels were computed from the output waveforms using the same procedure as that described in Section III-D. The higher number of minor differences between the input and output labels and higher correlation coefficients indicate better control of conditioning generation. Figure 4 illustrates a section of the results from the conditioning generation process.

The calculation procedure was as follows: A single input waveform, x , and an input label, c , from the dataset were selected. The input label, c , was a variable that comprised three precomputed values: *bright*, *warm*, and *rich*. From c , one evaluation label, c_e , was selected to evaluate the controllability

²<https://github.com/tsugumasa320/WavetableCVAE>

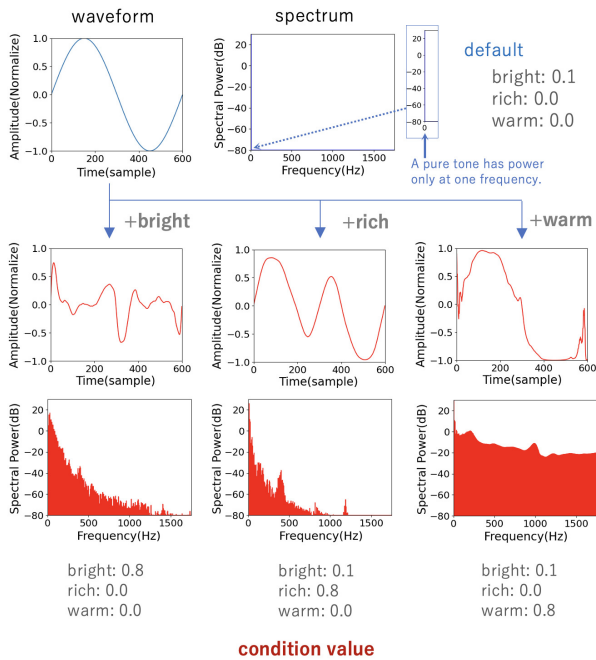


Fig. 4. Outcomes of conditional generation using sine waves as inputs: an analysis based on the Label characteristics. *bright* refers to the spectral centroid, *rich* denotes the energy ratio of the odd harmonics, and *warm* corresponds to the spectral density. The value associated with each label increases with the value of the conditioning.

of conditioning generation. The label c_e was set in order of 20 equally spaced values in the range of $[0, 1]$. Labels that were not used for evaluation were also needed for conditioning generation. The labels obtained from the input waveforms were calculated using the same procedure as in Section III-D.

The output waveform, \hat{x} , and the evaluation label, \hat{c}_e , were obtained from the input waveform, x , and the input label, c . This procedure was repeated for the number of divisions of c_e used as input (20 times in this research).

The transition between c_e and \hat{c}_e was stored as an array of 20 points. The MAE and Pearson correlation coefficients were calculated and evaluated from the stored arrays. These processes were repeated for WT, and the averages were obtained with the three parameters, *bright*, *warm*, and *rich*, for the entire dataset.

The value of MAE averaged over *bright*, *warm*, and *rich* were 0.19, 0.05, and 0.14, respectively. The *warm* value means the change is consistent with the label, whereas *bright* and *rich* remain challenging. The distribution of the training labels may have been biased, limiting the exposure of data to a sufficient range of labels for practical training.

The averaged values of the Pearson correlation coefficients were 0.97 for *bright*, 0.98 for *warm*, and 0.96 for *rich*. Strong positive correlations were confirmed for all parameters. This indicates that conditioning generation responds appropriately to the increase or decrease of each parameter.

C. Generation Time

The time required for generating WT was evaluated to ascertain the real-time performance capabilities of the proposed method. WT was generated 100 times, and the average

generation time was calculated to be approximately 2.7 ms. The evaluations were conducted on a MacBook Pro (13-inch, M1, 2020) with an M1 Chip (CPU), utilizing PyTorch Lightning³ 1.7.7. This notably brief generation time suggests the feasibility of real-time operation of our method using only the CPU without requiring a specialized GPU for deep learning.

VI. SUMMARY AND FUTURE ISSUES

In this research, we proposed a method for semantic timbre control of WT using CVAE, similar to audio effects, and demonstrated the possibility of improving the synthesizer programming problem. The following is the rationale for the proposed method:

This method shows that the reconstructed waveform can follow the input waveform by capturing the timbre features of the WT⁴. We have confirmed that the conditioning generation can be controlled according to the intention⁵. With regards to the generation time, the results reveal that real-time use is possible using only the CPU without needing a particular GPU for deep learning.

Future work will focus on developing a quantitative index for evaluating the reconstruction quality.

Furthermore, as detailed in Section V-B, the controllability assessment of the conditioning generation indicates significant MAE, particularly with *bright* and *rich* labels. The bias in training label distribution is likely the contributing factor, necessitating countermeasures. Results may be enhanced by augmenting the dataset size, altering label analysis, and normalization methods. Another strategy could involve adopting latent space and label feature separation mechanisms akin to those in Fader Networks [35].

As for real-time performance, model weight reduction methods such as model parameter quantization and branch pruning could further reduce the processing time of the proposed method.

REFERENCES

- [1] J. Shier, *The synthesizer programming problem: Improving the usability of sound synthesizers*, Master's thesis, University of Victoria, 2021.
- [2] G. Krekovic, "Insights in habits and attitudes regard-ing programming sound synthesizers: A quantitative study," in *Proceedings of the 16th Sound and Music Computing Conference*, 2019.
- [3] G. Kreković, "Deep Convolutional Oscillator: Synthesizing Waveforms from Timbral Descriptors," *Sound and Music Computing*, 2022.
- [4] D. L. Wessel, "Timbre space as a musical control structure," *Computer music journal*, pp. 45–52, 1979.
- [5] C. Hourdin, G. Charbonneau, and T. Moussa, "A sound-synthesis technique based on multidimensional scaling of spectra," *Computer Music Journal*, vol. 21, no. 2, pp. 56–68, 1997.

³<https://lightning.ai/docs/pytorch>

⁴Some of the results are shown in Figure 3

⁵Figure 4 shows how conditioning generation works on some labels

- [6] A. Horner, J. Beauchamp, and L. Haken, "Machine tongues xvi: Genetic algorithms and their application to fm matching synthesis," *Computer Music Journal*, vol. 17, no. 4, pp. 17–29, 1993.
- [7] M. Puckette, "Max at seventeen," *Computer Music Journal*, vol. 26, no. 4, pp. 31–43, 2002.
- [8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations*, 2014.
- [10] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [11] G. Kreković, "A concept of a wavetable oscillator based on a neural autoencoder," in *Proceedings of the 16th International Audio Mostly Conference*, 2021, pp. 240–243.
- [12] J. Hyrkas, "WaVAEtable Synthesis," *Proc. of the 15th International Symposium on CMMR*, p. 263, 2021.
- [13] N. Masuda and D. Saito, "Synthesizer sound matching with differentiable dsp.," in *International Society for Music Information Retrieval*, 2021, pp. 428–434.
- [14] N. Masuda and D. Saito, "Quality-diversity for synthesizer sound matching," *Journal of Information Processing*, vol. 31, pp. 220–228, 2023.
- [15] M. Chinen and N. Osaka, "Genesynth: Noise band-based genetic algorithm analysis/synthesis framework," in *Proceedings of the International Computer Music Conference*, 2007.
- [16] M. Macret, "Automatic tuning of the op-1 synthesizer using a multi-objective genetic algorithm," in *Proceedings of the Sound and Music Conference*, 2013.
- [17] A. Pošćić and G. Kreković, "Controlling a sound synthesizer using timbral attributes," in *Proceedings of the Sound and Music Computing Conference*, 2013, pp. 467–472.
- [18] G. Kreković and D. Petrinović, "Intelligent exploration of sound spaces using decision trees and evolutionary approach," in *Proceedings of the International Computer Music Conference joint with Sound and Music Computing Conference*, 2014, pp. 1263–1270.
- [19] S. Fasciani, "TSAM: A tool for analyzing, modeling, and mapping the timbre of sound synthesizers," in *Proceedings of 13th Sound and Music Computing Conference*, Aug. 2016, pp. 129–136.
- [20] F. Roche, T. Hueber, M. Garnier, S. Limier, and L. Girin, "Make that sound more metallic: Towards a perceptually relevant control of the timbre of synthesizer sounds using a variational autoencoder," *Transactions of the International Society for Music Information Retrieval*, vol. 4, pp. 52–66, 2021.
- [21] J. Engel, C. Resnick, A. Roberts, *et al.*, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*, PMLR, 2017, pp. 1068–1077.
- [22] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *International Conference on Learning Representations*, 2019.
- [23] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv preprint arXiv:2111.05011*, 2021.
- [24] P. Esling, N. Masuda, A. Bardet, R. Despres, and A. Chemla-Romeu-Santos, "Flow Synthesizer: Universal Audio Synthesizer Control with Normalizing Flows," *Applied Sciences*, vol. 10, no. 1, 2020.
- [25] R. Bristow-Johnson, "Wavetable synthesis 101, a fundamental perspective," in *Audio engineering society convention 101*, Audio Engineering Society, 1996.
- [26] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.
- [27] I. Higgins, L. Matthey, A. Pal, *et al.*, "Beta-vae: Learning basic visual concepts with a constrained variational representation," in *International conference on learning representations*, 2017.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [29] G. Kreković, A. Pošćić, and D. Petrinović, "An algorithm for controlling arbitrary sound synthesizers using adjectives," *Journal of New Music Research*, vol. 45, no. 4, pp. 375–390, 2016.
- [30] J. M. Grey and J. W. Gordon, "Perceptual effects of spectral modifications on musical timbres," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1493–1500, 1978.
- [31] K. K. Jensen, "Timbre models of musical sounds - from the model of one sound to the model of one instrument," in *Technical report / University of Copenhagen / Data-logisk institut*, 1999.
- [32] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidado project," *CUIDADO Ist Project Report*, vol. 54, no. 0, pp. 1–25, 2004.
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc Int Conf Learn Represent*, 2014.
- [34] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, "Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, pp. 240–250.
- [35] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader networks: Manipulating images by sliding attributes," *Advances in neural information processing systems*, vol. 30, 2017.