SYNTHCLONER: SYNTHESIZER PRESET CONVERSION VIA FACTORIZED CODEC WITH ADSR ENVELOPE CONTROL

Jeng-Yue Liu[1,2], Ting-Chao Hsu[1*], Yen-Tung Yeh[1], Li Su[2], Yi-Hsuan Yang[1]*

[1] National Taiwan University   [2] Academia Sinica

ABSTRACT

Electronic synthesizer sounds are controlled by presets, parameters settings that yield complex timbral characteristics and ADSR envelopes, making preset conversion particularly challenging. Recent approaches to timbre transfer often rely on spectral objectives or implicit style matching, offering limited control over envelope shaping. Moreover, public synthesizer datasets rarely provide diverse coverage of timbres and ADSR envelopes. To address these gaps, we present Synth-Cloner, a factorized codec model that disentangles audio into three attributes: ADSR envelope, timbre, and content. This separation enables expressive synthesizer preset conversion with independent control over these three attributes. Additionally, we introduce SynthCAT, a new synthesizer dataset with a task-specific rendering pipeline covering 250 timbres, 120 ADSR envelopes, and 100 MIDI sequences. Experiments show that SynthCloner outperforms baselines on both objective and subjective metrics, while enabling independent attribute control. The code, model checkpoint, and audio examples are available here[1].

Index Terms— synthesizer preset conversion, timbre transfer, ADSR envelope, dataset

1. INTRODUCTION

Electronic synthesizers have become essential tools in modern music production, offering vast flexibility in sound design. The sound of a synthesizer is defined by a preset, a parameter configuration of oscillators, filters, and envelopes, that jointly shape its sonic characteristics. Despite the creative possibilities this offers, crafting presets remains labor-intensive and requires expert knowledge [1].

To address this challenge, synthesizer preset conversion aims to convert a source preset to match a reference while preserving the content, thereby reducing manual parameter tuning and enabling efficient sound exploration. In this paper, we define the sonic character of a basic preset as comprising two aspects: timbre and ADSR envelope. Timbre denotes the *time-invariant* spectral characteristics shaped by oscillators and filters; for simplicity, we exclude LFO-induced vari-

---

* Equal contribution.

ations and additional envelope routing that introduce time-varying effects [2], while the ADSR envelope (Attack, Decay, Sustain, Release) governs the *time-varying* master amplitude trajectory. Together, these components produce the sound that range from percussive plucks to sustained leads. Consequently, effective preset conversion requires accurate modeling and control of both timbre and ADSR envelope.

Existing methods for timbre transfer [3, 4, 5, 6, 7, 8, 9] focus on acoustic settings and learn spectral or latent style mappings for reference-guided transfer, while guitar tone conversion [10] models the analog signal chain to reshape tone. Both paradigms primarily emphasize spectral similarity while leaving the input's ADSR envelope unchanged—an assumption valid for acoustic instruments but misaligned with synthesizer design. A related direction is synthesizer sound matching [1, 11, 12, 13, 14, 15, 16], which recovers parameters through differentiable or DSP-based models; however, preset conversion requires direct learning from audio for joint timbre and envelope control. Recent work has begun to consider envelopes more explicitly: Caspe *et al.* [17] predicted ADSR-like parameters to capture note onsets and offsets, while Masuda *et al.* [16] introduced a differentiable ADSR module. Yet these efforts focus on onset reproduction or differentiable control, rather than efficient preset conversion.

A further challenge is the lack of publicly available datasets that jointly span timbral and envelope diversity. NSynth [18] offers large-scale note-level samples with broad timbre coverage but mostly sustained notes and limited envelope variation, while Synth1B1 [19] provides massive synthetic audio with parameter metadata but no explicit envelope control. To address this gap, we introduce SynthCAT, a dataset built from systematic combinations of 250 timbres, 120 ADSR envelopes, and 100 MIDI sequences, capturing the full diversity of combinations.

Building on this resource, we present SynthCloner, a factorized codec model inspired by FACodec [20]. Synth-Cloner disentangles synthesizer audio into three attributes, content, timbre, and ADSR envelope, and integrates information perturbation [21] with supervised training [20] to enforce attribute separation. This enables faithful and controllable conversion across presets, explicitly modeling both spectral and temporal dynamics.

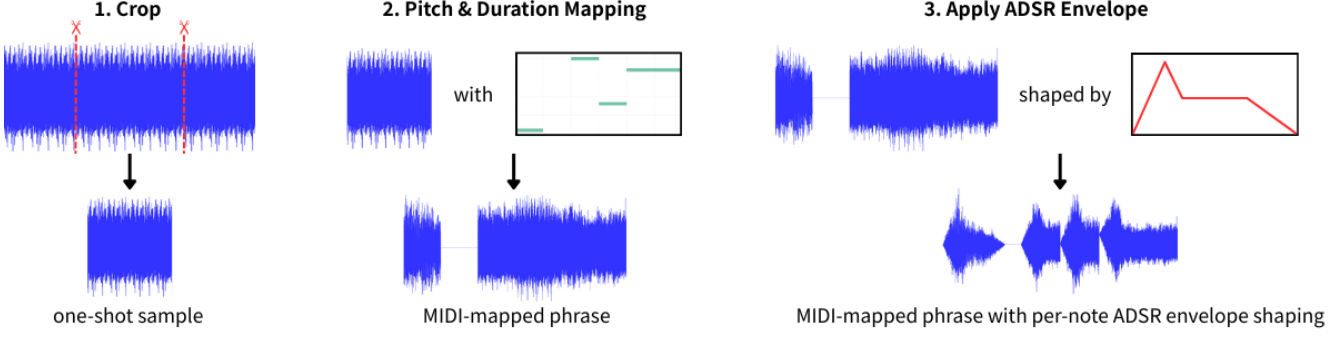The contributions of this paper are as follows:

Fig. 1. Data rendering pipeline of SynthCAT. A sustained-note segment is cropped, mapped to the MIDI sequence through pitch shifting and duration alignment, and then shaped with the ADSR envelope on a per-note basis to produce the final audio.

- We demonstrate empirically that explicit modeling of ADSR envelope control alongside timbre is essential for effective synthesizer preset conversion.

- We present SynthCloner, the first model to our knowledge that performs synthesizer preset conversion through factorization of timbre and ADSR envelope attributes.

- We introduce SynthCAT, a synthesizer dataset with 250 timbres, 120 ADSR envelopes, and 100 contents, providing a foundation for synthesizer preset conversion.

2. DATA

To construct SynthCAT, we use the `Serum`[2] synthesizer driven by MIDI notes, as Serum is one of the most widely used VST instruments in music production. Our goal is to render the same content under varied timbre and ADSR envelope settings. Isolating envelopes is non-trivial since many presets couple them with osicaltors, filters, or effects [22], leading to unwanted timbral changes. To mitigate this, we select presets from commercial packages[3] that generate sustained tones and render them as long notes. From each rendering, we extract a one-second one-shot sample from the flattest region of the waveform, quantified by the flatness score [23, 24]:

$$\text{Flatness}(x) = \frac{1}{1 + \text{Var}(\mu(x))}, \quad \mu(x) = \frac{1}{N} \sum_{n=1}^{N} x[n],$$

where $x[n]$ is the waveform of length N, $\mu(x)$ is its temporal mean, and $\text{Var}(\cdot)$ computes the variance across the segment. Segments with flatness above 0.95, indicating low variance in amplitude, are retained as distinct timbres. Each one-shot is then pitch-shifted and duration-aligned to MIDI sequences, and shaped with ADSR envelopes to render multi-note monophonic phrases. We generate 120 envelopes by uniformly sampling parameter ranges: Attack (10–100 ms),

Decay (50–300 ms), Hold (0–200 ms), Sustain (0.0–0.80), and Release (30–300 ms). Finally, the dataset is built from the Cartesian product of 250 timbres, 120 envelopes, and 100 MIDI files from the mono-midi-transposition dataset [25], yielding 3M monophonic audio samples (44.1 kHz, mono) with a total duration of about 2,500 hours.

3. PROPOSED APPROACH

3.1. The SynthCloner Framework

As illustrated in Fig. 2, SynthCloner is inspired by the FA-Codec [20] framework but incorporates two key extensions: ADSR envelope modeling and attribute-specific information perturbation, detailed in Section 3.2. The model decomposes an input audio $\mathbf{x}$ into three disentangled latent representations through separate processing paths, each with specific perturbations [21] during training.

First, the ADSR envelope path processes the perturbed input $\mathbf{x}_e$ by converting it to log-RMS value [26] and passing it through the ADSR envelope extractor which is a temporal multi-scale Conv-BiLSTM network to capture envelope dynamics [27], yielding the ADSR envelope embeddings $\mathbf{z}_e \in \mathbb{R}^{D \times T}$, where D denotes the embedding dimension and T the frame length. Second, the content path follows the FACodec [20] design, where the perturbed input $\mathbf{x}_c$ passes through the encoder and residual vector quantizer (RVQ) to produce the content embeddings $\mathbf{z}_c \in \mathbb{R}^{D \times T}$. Third, the timbre path, also adapted from FACodec [20], processes the perturbed input $\mathbf{x}_t$ through the shared-weight same encoder and a Conformer-based [20, 28] timbre extractor with global mean pooling, generating the timbre embedding $\mathbf{z}_t \in \mathbb{R}^D$ that captures global spectral characteristics. During reconstruction, $\mathbf{z}_e$ is additively combined with $\mathbf{z}_c$, then modulated by $\mathbf{z}_t$ through conditional layer normalization [29], before being decoded back to the waveform. For inference, the model enables synthesizer preset conversion by using a reference audio as $\mathbf{x}_e$ and $\mathbf{x}_t$ to provide target timbre and ADSR envelope while preserving the original content.
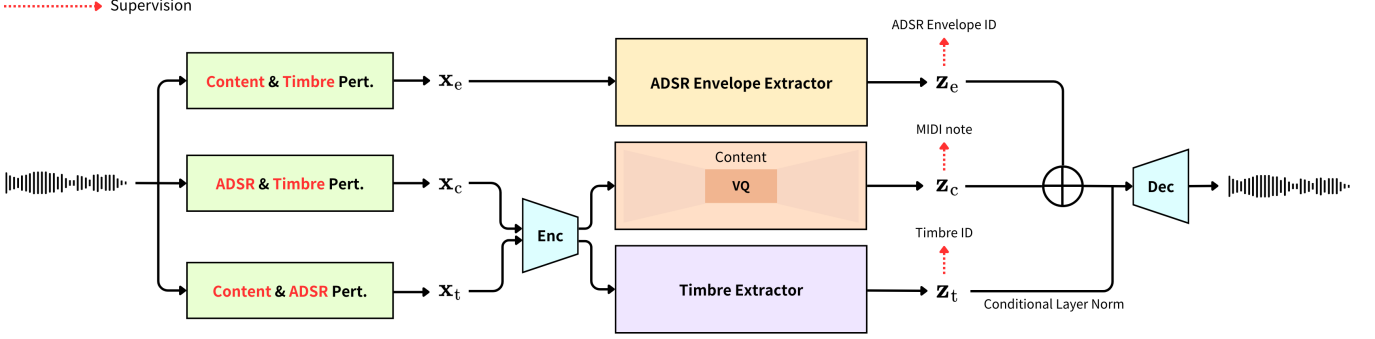
---

[2] https://xferrecords.com/products/serum-2
[3] https://theproducerschool.com/

Fig. 2. SynthCloner model architecture. To perform synthesizer preset conversion, replace $\mathbf{x}_e$ and $\mathbf{x}_t$ with the reference audio.

3.2. Attribute Disentanglement

While the three-path architecture provides a structural basis for attribute separation, it does not guarantee the disentanglement. We introduce attribute-specific information perturbation, inspired by NANSY [21], to ensure that $\mathbf{z}_e$, $\mathbf{z}_c$, and $\mathbf{z}_t$ capture only their target attributes. During training, non-target attributes in the input are perturbed while the target attribute is preserved, forcing each encoder to produce embeddings invariant to irrelevant variations. SynthCAT, with its exhaustive combinations of timbre, ADSR envelope, and content, enables such perturbations. For example, when training the ADSR envelope encoder, given an input $\mathbf{x}$ with envelope e_0, content c_0, and timbre t_0, we construct a perturbed input $\mathbf{x}_e$ that shares e_0 but differs in content c_1 and timbre t_1, ensuring $\mathbf{z}_e$ remains invariant to non-envelope factors. Analogous perturbations are applied to the content and timbre paths. Following FACodec [20], we further employ attribute-specific auxiliary tasks for disentanglement encouragement. For content embeddings $\mathbf{z}_c$, we use MIDI-derived pitch labels for frame-level supervision. For ADSR envelope and timbre embeddings ($\mathbf{z}_e$, $\mathbf{z}_t$), we apply categorical classification over 120 envelope IDs and 250 timbre IDs, respectively.

4. EXPERIMENT SETUP

4.1. Training Setup

Dataset. For evaluation, we hold out 250 seen timbres along with 20 unseen envelopes and 10 unseen MIDI sequences, yielding 50,000 testing samples. Each testing sample serves as a source, with 10 reference samples randomly selected; the corresponding ground-truth outputs are available for direct comparison.

Implementation Details. SynthCloner is implemented in PyTorch and trained on a single NVIDIA RTX 6000 Ada GPU for 400k steps. The model processes 1-second audio segments at 44.1 kHz using the AdamW optimizer [30] with an initial learning rate of 10^{-4} and an exponential decay rate of 0.999996. The batch size is 8, and the RVQ module uses a 1024-entry codebook with 8 quantization layers. We use same loss functions in FACodec [20]. For reconstruction, we apply a multi-scale mel-spectrogram loss with 7 scales using FFT window lengths of [32, 64, 128, 256, 512, 1024, 2048] paired with mel bins of [5, 10, 20, 40, 80, 160, 320], respectively. For each scale, the hop length is set to window_length/4. The total training objective is a weighted sum of multi-scale mel spectrogram loss $\lambda_{\text{mel}}\mathcal{L}_{\text{mel}}$, feature matching loss $\lambda_{\text{feat}}\mathcal{L}_{\text{feat}}$, adversarial loss $\lambda_{\text{adv}}\mathcal{L}_{\text{adv}}$, commitment loss [31] $\lambda_{\text{commit}}\mathcal{L}_{\text{commit}}$, codebook loss [31] $\lambda_{\text{codebook}}\mathcal{L}_{\text{codebook}}$, and classification losses for timbre, content, and ADSR envelope ($\lambda_{\text{timbre}}\mathcal{L}_{\text{timbre}} + \lambda_{\text{content}}\mathcal{L}_{\text{content}} + \lambda_{\text{adsr}}\mathcal{L}_{\text{adsr}}$). We set $\lambda_{\text{mel}} = 15.0$, $\lambda_{\text{feat}} = 2.0$, $\lambda_{\text{adv}} = 1.0$, $\lambda_{\text{commit}} = 0.25$, $\lambda_{\text{codebook}} = 1.0$, and $\lambda_{\text{timbre}} = \lambda_{\text{content}} = \lambda_{\text{adsr}} = 5.0$.

4.2. Evaluation Setting

Baseline Models. Since no prior work addresses synthesizer preset conversion directly, we compare SynthCloner against two many-to-many timbre transfer models: (1) SS-VAE [4], a semi-supervised variational autoencoder for controllable audio generation, and (2) Control Transfer Diffusion (CTD) [6], a diffusion-based method for conditional timbre transfer. We further evaluate a variant of our model without the ADSR envelope path, keeping the ADSR envelope fixed during perturbations in other paths to assess the impact of explicit ADSR modeling. Both the proposed and baseline models are trained on our SynthCAT dataset from scratch.

Objective Metrics. We evaluate preset conversion with three metrics. Spectral fidelity is measured by the multi-scale STFT loss (MSTFT) [32] from audiotools [33] , which computes L1 distances over magnitude and log-magnitude spectra using STFT windows of 2048 and 512. ADSR envelope accuracy is assessed by the log-RMS distance (LRMSD), the L1 difference between log-RMS energy contours of the prediction and reference. Content accuracy is measured by the F0 root mean square error (F0RMSE), with pitch contours extracted via TorchCrepe [34].

Subjective Metrics. We assess perceptual quality through Mean Opinion Score (MOS) tests on three dimensions: tim-

	Objective Metrics			Subjective Metrics		
	MSTFT↓	LRMSD↓	F0RMSE↓	TMOS↑	ADSRMOS↑	CMOS↑
Ground Truth	–	–	–	4.08	3.96	4.25
SS-VAE [4]	7.22	0.92	641.62	2.20	2.25	3.41
CTD [6]	5.69	0.89	583.01	2.34	2.48	1.86
SynthCloner (ours)	**3.00**	**0.17**	**20.64**	**3.91**	**3.94**	**4.11**
– w/o ADSR envelope path	3.84	0.42	29.04	3.09	2.40	3.76

Table 1. Objective and subjective results for synthesizer preset conversion.

bre similarity (**TMOS**) and ADSR envelope similarity (**ADSRMOS**), measured between each prediction and its reference, and content similarity (**CMOS**), measured between each prediction and its source. Listeners rated samples on a five-point Likert scale from 1 (bad") to 5 (excellent"). Four source–reference pairs were randomly selected; for each pair, all models produced 4 predictions plus the ground truth (5 samples total). Twenty subjects rated all samples, and MOS with 95% confidence intervals were reported.

5. EXPERIMENT RESULTS

5.1. Synthesizer Preset Conversion

As shown in Table 1, SynthCloner achieves the best performance across both objective and subjective metrics. The low MSTFT confirms that the predicted outputs closely match the ground-truth signals in spectral structure, demonstrating high fidelity in preset conversion. ADSR envelope accuracy follows the same trend: LRMSD remains low, consistent with faithful reproduction of target ADSR envelopes. Finally, content preservation is reflected by a low F0RMSE, suggesting that pitch trajectories are stable when timbre and ADSR are altered. Compared to the baselines, SS-VAE and CTD exhibit higher MSTFT and LRMSD and larger F0 deviations, pointing to pitch instability when ADSR envelope and timbre are manipulated jointly. The subjective results corroborate the objective metrics. SynthCloner achieves the highest MOS across all dimensions, approaching ground truth, showing that listeners perceived its outputs as highly similar to the reference in timbre and ADSR envelope while preserving content.

When trained without the ADSR envelope path, LRMSD rises notably and MSTFT increases, confirming that envelope dynamics cannot be captured without explicit modeling and indicating degraded spectral fidelity. However, F0RMSE remains relatively close to the full model, showing pitch content is largely unaffected. The subjective metrics reinforce this: ADSRMOS drops dramatically and TMOS decreases substantially without the envelope path, while CMOS shows a more modest decline, consistent with preserved pitch content. These findings underscore that explicit ADSR envelope modeling is crucial for effective synthesizer preset conver-

	MSTFT ↓	LRMSD ↓	F0RMSE ↓
Proposed	**3.00**	**0.17**	**20.64**
w/o timbre conv.	5.97	0.19	24.54
w/o ADSR conv.	4.15	0.39	24.06

Table 2. Objective result of independent attribute control.

sion, significantly impacting spectral fidelity and perceptual quality while maintaining content integrity.

5.2. Ablation Study

In this section, we evaluate the independent attribute control of SynthCloner by three settings: (1) **Proposed**, which performs both timbre and ADSR envelope conversion, (2) **w/o timbre conv.**, which uses the input timbre instead of the reference, and (3) **w/o ADSR conv.**, which uses the input ADSR envelope instead of the reference. Relative to Proposed, **w/o timbre conv.** yields higher MSTFT, indicating that timbre conversion is crucial for spectral fidelity, while its LRMSD is only slightly worse, suggesting that envelope transfer remains unchanged. In contrast, **w/o ADSR conv.** shows much higher LRMSD than Proposed, demonstrating that envelope matching fails without explicit ADSR conversion. For both partial conversions, F0RMSE is slightly worse than Proposed, indicating that pitch content is largely preserved but less stable when either attribute is missing.

6. CONCLUSION

We presented SynthCloner, the first model for synthesizer preset conversion that disentangles ADSR envelope, timbre, and content. We also introduced SynthCAT, a dataset with a rendering pipeline tailored for this task. Experiments show that SynthCloner outperforms baselines across all metrics, and ablations underscore the necessity of explicit ADSR modeling. Future work will aim to improve generalization to unseen timbres and extend the model to more complex preset behaviors, including LFO modulations, additional envelope routing, and coupled effects.

7. REFERENCES

[1] Ben Hayes, Charalampos Saitis, and György Fazekas, "Audio synthesizer inversion in symmetric parameter spaces with approximately equivariant flow matching," in *Proc. ISMIR*, 2025.

[2] Mark Vail, *The Synthesizer: A Comprehensive Guide to Understanding, Programming, Playing, and Recording the Ultimate Electronic Music Instrument*, Oxford University Press, New York, 2014.

[3] Michele Mancusi, Yurii Halychanskyi, Kin Wai Cheuk, Eloi Moliner, et al., "Latent diffusion bridges for unsupervised musical audio timbre transfer," in *Proc. ICASSP*, 2025.

[4] Ondrej Cifka, Alexey Ozerov, Umut Simsekli, and Gael Richard, "Self-supervised vq-vae for one-shot music style transfer," in *Proc. ICASSP*, 2021.

[5] Sifei Li, Yuxin Zhang, Fan Tang, Chongyang Ma, Weiming dong, and Changsheng Xu, "Music style transfer with time-varying inversion of diffusion models," in *Proc. AAAI*, 2024.

[6] Nils Demerlé, Philippe Esling, Guillaume Doras, and David Genova, "Combining audio control and style transfer using latent diffusion," in *Proc. ISMIR*, 2024.

[7] Yuxuan Wu, Yifan He, Xinlu Liu, Yi Wang, and Roger B. Dannenberg, "Transplayer: Timbre style transfer with flexible timbre control," in *Proc. ICASSP*, 2023.

[8] Luca Comanducci, Fabio Antonacci, and Augusto Sarti, "Timbre transfer using image-to-image denoising diffusion implicit models," in *Proc. ISMIR*, 2023.

[9] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, et al., "Timbretron: A wavenet(cyclegan(cqt(audio))) pipeline for musical timbre transfer," in *Proc. ICLR*, 2023.

[10] Yu-Hua Chen, Yen-Tung Yeh, Yuan-Chiao Cheng, et al., "Towards zero-shot amplifier modeling: One-to-many amplifier modeling via tone embedding control," in *Proc. ISMIR*, 2024.

[11] Fred Bruford, Frederik Blang, and Shahan Nercessian, "Synthesizer sound matching using audio spectrogram transformers," in *Proc. DAFx*, 2024.

[12] Zui Chen, Yansen Jing, Shengcheng Yuan, Yifei Xu, Jian Wu, and Hang Zhao, "Sound2synth: Interpreting sound via fm synthesizer parameters estimation," in *Proc. IJCAI*, 2022.

[13] Andrew Horner, James Beauchamp, and Lippold Haken, "Machine tongues xvi: Genetic algorithms and their application to fm matching synthesis," *Computer Music Journal*, 1993.

[14] Matthew John Yee-King, Leon Fedden, and Mark d'Inverno, "Automatic programming of vst sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2018.

[15] Oren Barkan, David Tsiris, Ori Katz, and Noam Koenigstein, "Inversynth: Deep estimation of synthesizer parameter configurations from audio signals," in *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 2019.

[16] Naotake Masuda and Daisuke Saito, "Improving semi-supervised differentiable synthesizer sound matching for practical applications," in *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 2023.

[17] Franco Caspe, Andrew McPherson, and Mark Sandler, "FM Tone Transfer with Envelope Learning," in *Proc. International Audio Mostly Conference*, 2023.

[18] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, et al., "Neural audio synthesis of musical notes with wavenet autoencoders," in *Proc. ICML*, 2017.

[19] Joseph Turian, Jordie Shier, George Tzanetakis, Kirk McNally, and Max Henry, "One billion audio sounds from gpu-enabled modular synthesis," in *Proc. DAFx*, 2021.

[20] Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, et al., "Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," in *Proc. ICML*, 2024.

[21] Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Hwan Lee, et al., "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," in *Proc. NeurIPS*, 2021.

[22] Christopher Mitcheltree, Christian J. Steinmetz, Marco Comunità, and Joshua D. Reiss, "Modulation extraction for lfo-driven audio effects," in *Proc. DAFx*, 2023.

[23] A. Gray and J. Markel, "A spectral-flatness measure for studying the autocorrelation method of linear prediction of speech analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1974.

[24] Yanna Ma and Akinori Nishihara, "Efficient voice activity detection algorithm using long-term spectral flatness measure," *EURASIP Journal on Audio, Speech, and Music Processing*, 2013.

[25] Sebastian Garcia-Valencia, Alejandro Betancourt, et al., "Sequence generation using deep recurrent networks and embeddings: A study case in music," *ArXiv e-prints*, 2020.

[26] Marcelo Caetano and Xavier Rodet, "Improved estimation of the amplitude envelope of time-domain signals using true envelope cepstral smoothing," in *Proc. ICASSP*, 2011.

[27] Nicholas Wilkinson and Thomas Niesler, "A hybrid cnn-bilstm voice activity detector," in *Proc. ICASSP*, 2021.

[28] Yang Zhang, Zhiqiang Lv, Haibin Wu, Shanshan Zhang, et al., "Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification," *Proc. Interspeech*, 2022.

[29] Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, Sheng Zhao, and Tie-Yan Liu, "Adaspeech: Adaptive text to speech for custom voice," in *Proc. ICLR*, 2021.

[30] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[31] Aaron Van Den Oord, Oriol Vinyals, et al., "Neural discrete representation learning," *Advances in neural information processing systems*, 2017.

[32] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts, "DDSP: Differentiable digital signal processing," in *Proc. ICLR*, 2020.

[33] Inc. Descript, "audiotools: Object-oriented handling of audio data, with gpu-powered augmentations, and more," https://github.com/descriptinc/audiotools, 2024.

[34] Max Morrison, "torchcrepe," https://github.com/maxrmorrison/torchcrepe, October 2023.