Published at the IEEE CAI Conference, May 5-7, 2025 Santa Clara, CA, USA

DiffCraft: A Modular Differentiable Framework for Music Synthesis with Timbre Transfer

Venkat Suprabath Bitra
Dept. of Computer Science
Columbia University New York, US
vsb2127@columbia.edu

Homayoon Beigi
Dept. of Mechanical Engineering and Dept. of Electrical Engineering
Columbia University
Recognition Technologies, Inc.
New York, US
hb87@columbia.edu

***Abstract*—This paper presents a novel autoencoder-based framework for music synthesis and timbre transfer, addressing limitations of traditional Differentiable Digital Signal Processing (DDSP). By integrating a Variational Autoencoder (VAE) based Audio Encoder with a Mamba-enhanced DDSP Decoder, the framework extracts fundamental frequency (F_0) and loudness (L) directly from waveforms, eliminating reliance on pre-extracted features. The use of Piano Complex Cepstral Coefficients (PCCC) with VAE allowed learning a smooth and rich feature space, enabling flexible audio manipulation. Bidirectional Mamba demonstrated improved performance over the baseline DDSP model. The system achieves high-quality audio generation and efficient timbre transfer. The proposed framework is versatile, offering applications in music transcription, timbre transfer, and multi-instrument synthesis.**

***Index Terms*—ddsp, music, autoencoder, mamba, timbre transfer**

I. Introduction

In the domain of machine learning based signal processing, music synthesis and source separation are complex tasks for a model to learn directly from audio sources. Though it is challenging, it has many applications in melody extraction, pitch estimation, music transcription, music remixing, karaoke etc. For this problem, Differentiable Digital Signal Processing (DDSP) is an evolving framework that merges neural networks with traditional signal processing techniques. It is an end-to-end framework with integration of domain knowledge from signal processing by leveraging the learning capabilities of deep learning models and interpretability of DSP. Moreover, DDSP leverages prior knowledge about audio signals to generalize across various musical instruments, even when faced with limited training data, a challenge for traditional approaches.

The DDSP framework uses fundamental frequency extracted from audio samples using CREPE, a pitch tracker that employs a deep convolutional neural network. However, this approach of DDSP has limitations. It relies on pre-extracted features, which requires preprocessing and may not generalize well. It struggles with multi-instrument scenarios and complex timbre-temporal dynamics interactions, especially with limited training data. The fixed structure of DDSP models also restricts flexibility for tasks like timbre transfer or instrument-specific synthesis without significant retraining.

In this context, integration of an autoencoder-based synthesis approach could effectively address some of the limitations of DDSP. It learns representations of fundamental frequency (F_0) and loudness (L) directly from input waveforms, eliminating dependency on external extractors. This approach simplifies preprocessing and creates an instrument-specific embedding space. The autoencoder consolidates F_0 extraction and loudness estimation into one framework. It can generate novel F_0 and loudness distributions for the instrument, enabling applications like synthetic dataset generation, timbre-based audio synthesis, and instrument-specific audio composition. This method provides a flexible solution for music generation and audio processing.

Therefore, this paper presents an autoencoder-based architecture that addresses challenges in music synthesis, adaptive timbre synthesis, and end-to-end training for diverse audio tasks. The proposed architecture combines a novel Audio Encoder with an enhanced DDSP Decoder to create a modular and differentiable framework. The study explores two distinct approaches of Audio Encoder, i.e., first a Dual Encoding method that leverages both time-domain and frequency-domain information, and the second encoding strategy is based on using Piano Complex Cepstral Coefficients (PCCC) to separate F_0 and Loudness for targeted manipulation of audio characteristics. It also addresses improvement of the DDSP Decoder using Mamba, an advanced recurrent architecture along with bidirectional models. The Embedding space analysis to understand the distinct patterns in F_0 and loudness representations is presented. The main objective of this work is to develop a novel audio synthesis framework that combines VAE-based encoding with Mamba-enhanced DDSP synthesis for high-quality sound generation and timbre transfer. It aims to disentangle fundamental frequency (F_0) and loudness information in the latent space allowing more interpretable and controlled audio synthesis. The proposed modular architecture is versatile for various audio syntheses and processing, makes it attractive and finds various applications.

II. Literature Review

The use of DDSP encompasses audio synthesis, MSS, pitch and timbre control, among other applications. Recently, DDSP has demonstrated significant progress in fields like differen-

tiable wavetable synthesis [1], style transfer of audio effects [2], bandwidth extension of music signals [3], modulation synthesis for sound matching [4], attention-based audio speaker tracking [5], rendering and identification of impact sounds [6], as well as acoustically guided sound effects [7]. A multi-pitch estimator has been used in an end-to-end training model for differentiable singing voice separation, working well with limited data [8]. Additionally, music source separation using a parametric source model was introduced with a differentiable source-filter model [9]. This model allows reconstruction of sound mixtures by estimating source model parameters based on fundamental frequencies.

Recent studies have explored the application of Mamba, a state space model-based architecture, in audio processing tasks. Audio Mamba demonstrated comparable or better performance than Audio Spectrogram Transformers in audio classification across multiple benchmarks [10]. DeFT-Mamba showed significant improvements in universal sound separation and polyphonic audio classification [11]. Additionally, Mamba exhibited competitive performance in various speech applications, including ASR, text-to-speech, and speech summarization, while demonstrating efficiency in processing long-form speech [12]. Variational Autoencoders (VAEs) are generative models widely used in audio processing, offering advantages like smooth interpolation between samples and attribute disentanglement. Recent advancements have improved VAEs' performance in handling long-term dependencies and generating high-quality waveforms efficiently. The MusicVAE model introduced a hierarchical decoder for better long-term structure in musical sequences, enhancing sampling, interpolation, and reconstruction [13]. The RAVE model further advanced VAE applications, enabling fast, high-quality waveform synthesis at 48kHz, running 20 times faster than real-time on standard CPUs [14]. However, the exploration of VAE-based audio latent space for F_0 and loudness extraction, particularly when combined with state-of-the-art Mamba architectures in both unidirectional and bidirectional configurations for DDSP synthesis, remains unexplored.

III. Methodology

The audio processing architecture uses an autoencoder-based framework focusing on estimation of F_0 and L, as well as audio synthesis. Figure 1 illustrates the implemented autoencoder with the DDSP Decoder and Synthesizer blocks.

It has an Audio Encoder that maps the input audio into a latent vector z. This latent representation is parameterized using two distinct approaches:

- Differential (Deterministic) Parameterization: In this, the latent vector is computed as the direct difference between two feature embeddings, resulting in a standard autoencoder representation.
- Variational Autoencoder (VAE): Here, the latent vector z is defined by a mean (μ) and log variance $(log(\sigma^2))$. This probabilistic representation allows the model to learn a smooth feature space, potentially capturing more nuanced audio characteristics.

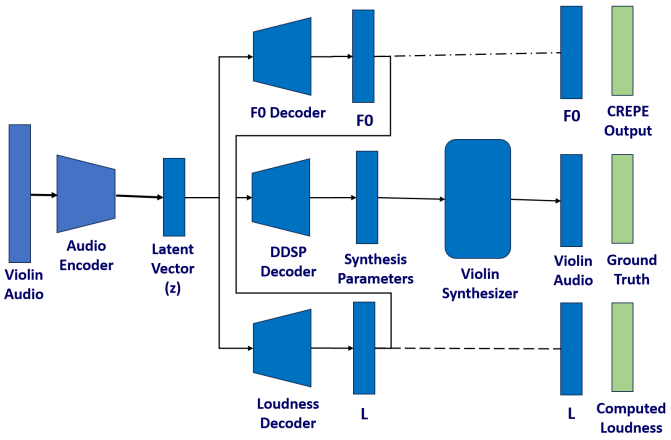

Fig. 1. Proposed Architecture diagram.

The obtained latent vector z is then processed through two task-specific decoders for the input audio, i.e., F_0 Decoder which predicts the fundamental frequency and L Decoder estimates the loudness. The outputs of these decoders are used to generate synthesis parameters for the DDSP Decoder. For example, the parameters drive the Violin Synthesizer, enabling reconstruction of violin audio. It is compared against the ground truth to refine the model. Additionally, F_0 and L are also validated with ground-truth references. F_0 is validated using CREPE (a pitch estimation algorithm). L is validated using computed power spectrum. This architecture allows efficient audio processing for accurate estimation of F_0, L, and high-quality synthesis of audio.

A. DDSP Decoder

The DDSP Decoder architecture inspired by the work of Engel et al. (2020), with several modifications and enhancements, it is used in this study to meet specific requirements. It generates sound synthesis from F_0 and L, which are vital for describing the harmonic and dynamic characteristics of audio. To establish a baseline, we initially trained the decoder as outlined by Engel et al., using a Gated Recurrent Unit (GRU) to generate synthesis parameters from F_0 and L. In our implementation, the z-encoder is removed to simplify the architecture. This adjustment streamlines the computation and reducing training complexity for capturing the essential characteristics of audio from F_0 and L. The same has been showcased in Figure 2.

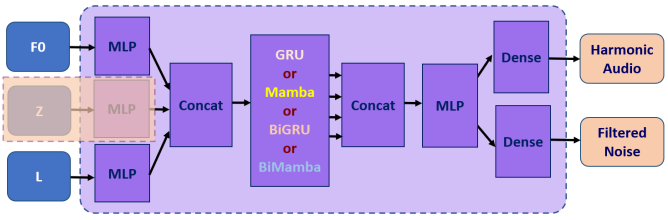

Fig. 2. Modified DDSP Architecture

To enhance the decoder performance, we replaced the GRU in the original DDSP model with a more recent recurrent architecture Mamba, which has shown to better capture temporal dependencies and leverage contextual information within text sequences. Unlike traditional recurrent neural networks (RNNs) or Long Short-Term Memory (LSTM) networks, Mamba employs a novel architecture that allows it to process sequences with linear time complexity while maintaining the ability to model complex temporal relationships. Hence, Mamba model well suited for this task due to its ability to efficiently capture long-range dependencies in sequential data.

For further improvements we experimented with bidirectional architectures, namely Bidirectional GRU (Bi-GRU) and Bidirectional Mamba (BiMamba). Bidirectional models incorporate contextual information from both past and future frames, particularly advantageous in audio synthesis tasks [10], where F_0 and L are known for the entire duration of a clip. The temporal nature of audio signals necessitates capturing relationships across both past and future frames to accurately reproduce dynamics and nuances. For example:

- In a musical passage with a crescendo, knowing both the start and peak of the loudness change ensures a smooth and realistic transition.
- Synthesizing vibrato effects requires knowledge of the complete oscillatory pattern in F_0, including its beginning and end, to accurately replicate it.

Bidirectional models are crucial for tasks such as pitch transitions, dynamic swells, or pauses, where future frames contain critical cues for generating accurate synthesis parameters.

B. Audio Encoder

The Audio Encoder is a critical component in our system, responsible for transforming raw audio signals into robust latent representations. We have implemented two distinct approaches to achieve this goal. The general architecture design for both these approaches is provided in Figure 3.

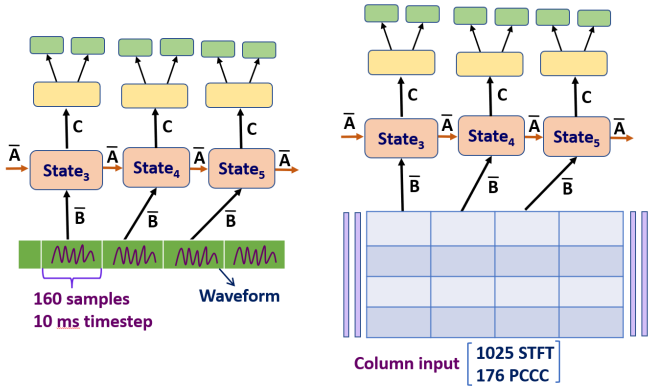

Fig. 3. Proposed Waveform, STFT and PCCC encoders

1) Dual Encoding Approach: This Approach combines waveform and spectrogram encoding to capture comprehensive audio features. The waveform encoder processes 4-second audio clips (64000 samples at 16 kHz) into 400 non-overlapping 10 ms snippets, each transformed into a 160-dimensional embedding vector by reshaping. This waveform adjusted in a sequence is processed by a Mamba model. The STFT encoder applies Short-Time Fourier Transform with 2048 bins, a 512-sample Hanning window, and 160-sample hop length. The resulting 1025x401 is processed to generate a log-magnitude spectrogram which is processed by another Mamba model. The final representation integrates outputs from both encoders through concatenation. This approach captures both fine-grained temporal dynamics and broader spectral characteristics, creating a robust latent representation for various audio processing tasks.

2) Separate F_0 and Loudness Encoding: This approach learns distinct representations for fundamental frequency (F_0) and loudness. It is based on Piano Complex Cepstral Coefficients (PCCC) for frequency encoding and a waveform encoder for loudness information. The PCCC Encoder extracts frequency information, including harmonics and overtones, focusing on piano-like sounds.

The PCCC can be described with the following equations:

- Complex filterbank application to STFT spectrum:

$$S(k) = \sum_{f=0}^{N-1} X(f) H_k(f)$$

where $X(f)$ is the complex STFT of the input signal, $H_k(f)$ is the k-th complex triangular filter, and N is the number of frequency bins. The filterbank H was designed using triangular filters centered at piano key frequencies, with overlaps defined by neighboring keys.

- Complex Log Spectrum Computation:

$$L(k) = log(S(k) + \epsilon)$$

where ϵ is a small constant to avoid log(0).

- Discrete Cosine Transform (DCT) and Discrete Sine Transform (DST) application to real and imaginary parts respectively:

$$C(m) = \sum_{k=0}^{K-1} \text{Re}(L(k)) \cos\left(\frac{\pi m(k+0.5)}{K}\right)$$

$$S(m) = \sum_{k=1}^{K} \text{Im}(L(k)) \sin\left(\frac{\pi m k}{K}\right)$$

where K is the number of filters (88 piano keys), m is the cepstral coefficient index, and Re and Im denote real and imaginary parts respectively.

The final PCCC features are formed by concatenating the coefficients from both transforms.

$$PCCC = [C(0), ..., C(M-1), S(0), ..., S(M-1)]$$

Loudness information is extracted directly from the audio waveform using a separate encoder similar to the previously discussed waveform encoder. This approach allows for targeted manipulation and analysis of audio characteristics.

C. F_0 and L Decoder

The F_0 and L Decoders are dense neural networks with the activation function $tanh$. The F_0 decoder is designed to estimate pitch values in cents, a logarithmic unit representing musical intervals relative to a reference pitch f_{ref} (10 Hz in this case). The relationship between frequency f in Hz and cents is given by: $c(f) = 1200 \cdot \log_2 \frac{f}{f_{\text{ref}}}$ The decoder outputs a 435-dimensional vector, with each dimension representing a 20-cent frequency bin covering six octaves from A0 (27.50 Hz) to C8 (4186.01 Hz). The pitch estimate $\hat{c}$ is calculated as a weighted average of the associated pitches c_i according to the output $\hat{y}$: $\hat{c} = \frac{\sum_{i=1}^{435} \hat{y}_i c_i}{\sum_{i=1}^{435} \hat{y}_i}$ The frequency estimate in Hz is then derived as: $\hat{f} = f_{ref} \cdot 2^{\hat{c}/1200}$ To train the model, the target outputs are 435-dimensional vectors which represents the probability distribution of the fundamental frequency and sums up to 1. To soften the penalty for close predictions, the target is Gaussian-blurred in frequency with a standard deviation of 25 cents: $y_i = \exp\left(-\frac{(c_i - c_{\text{true}})^2}{2 \cdot 25^2}\right)$ This soft-penalty approach allows for better pitch estimation and training.

The loudness decoder outputs a single value representing the loudness in decibels (dB). The decoder solves a regression problem using the sum of mean squared error (MSE) and mean absolute error (MAE) loss between the predicted and ground truth loudness values. The ground truth loudness is computed as the log-magnitude of the power spectrum averaged across frequency bins.

IV. Results

A. Dataset

Our experiments used two datasets: the MDB-stem-synth dataset and a custom dataset from Bach's Violin Partita No. 1. The MDB-stem-synth dataset is a benchmark for music information retrieval and synthesis tasks, containing 230 tracks of synthesized multi-instrument recordings. We focused on 14 violin tracks, which is about 50 minutes of effective audio after removing extended silences. We applied a 100 ms silence threshold and resampled the audio to 16 kHz.

The custom dataset from Bach's Violin Partita No. 1 provided real-world violin recordings. It includes five movements performed by John Garner, offering about 13 minutes of solo violin music. These recordings were already at 16 kHz, matching our model's requirements. This combination of synthesized and real-world data allowed for a comprehensive evaluation of our model's performance.

B. Audio Encoder

We evaluated two primary encoding approaches: the Dual Encoding Model and the separate F_0 and L Encoders. Both approaches were trained on the Bach Violin Partita dataset, providing insights into their respective performances and limitations. The training process consisted of 300 epochs with a step learning rate decay, scaling the learning rate by 0.995 every 2 steps, starting from a base learning rate of 2×10^{-4}.

For the Dual Encoding Model, the Differential Autoencoder (AE) approach demonstrated robust convergence. The model achieved a Multiscale Spectral Loss (MSS) loss of 5.936, indicating good reconstruction of the audio. The F_0 loss, representing the pitch or fundamental frequency error, was 2.686, reflecting the model's accuracy in predicting frequency components. Notably, the Loudness loss was 0.046, suggesting excellent performance in capturing the audio's loudness characteristics. These results indicate that the Differential AE approach effectively captured essential audio features. In contrast, the Variational Autoencoder (VAE) approach encountered significant challenges. The model failed to converge on the dataset due to posterior collapse. Despite implementing various techniques from the literature to mitigate this issue, the model's performance remained poor. The learned F_0 exhibited substantial deviation from the ground truth, resulting in exceptionally high loss values. Given these limitations, we did not pursue further exploration of the Dual Encoding Model on the MDB-stem-synth dataset.

The separate F_0 and loudness models demonstrated strong convergence during training. It exhibited superior performance and better convergence characteristics. Following the promising results from this model on the Bach Violin Partita dataset, we extended our evaluation to the more diverse MDB-stem-synth dataset. We compared the performance of two encoding approaches: Differential Encoding and Variational Autoencoder (VAE) Encoding, by training the different decoders, i.e., F_0 Decoder, Loudness Decoder and DDSP Decoder.

Loss		**Differential Encoding**	**VAE Encoding**
F_0	**Mean**	3.542 $\pm$ 0.134	3.341 $\pm$ 0.115
	SD	3.705	3.184
Loudness	**Mean**	0.072 $\pm$ 0.002	0.072 $\pm$ 0.002
	SD	0.054	0.054
MSS	**Mean**	5.796 $\pm$ 0.032	5.796 $\pm$ 0.033
	SD	0.900	0.914

Both Differential and VAE Encoding schemes have shown similar performance across metrics. VAE has lower F_0 loss (3.341 $\pm$ 0.115) as compared to that of differential encoding (3.542 $\pm$ 0.134). Both methods demonstrate identical Loudness loss (0.072 $\pm$ 0.002) and MSS loss (5.796 $\pm$ 0.03). These results indicate comparable effectiveness on the MDB-stem-synth dataset, with VAE showing slight advantage in F_0 prediction.

Since, we observe that both Differential and Autoencoder approaches are provided good results, we extended the analysis to explore the properties of the VAE embedding space. Figure 4 presents the t-SNE projections of the F_0 and Loudness embeddings learned by the VAE model on the MDB-stem-synth dataset.

The distinct patterns observed in these embeddings reflect the inherent nature of musical attributes. The loudness exhibited a smooth, continuous distribution in the embedding space (right), while the fundamental frequencies (F_0) form discrete clusters (left). This clustering behavior in F_0 embeddings attributed to the discrete nature of musical notes, ranging from A0 to C8 on the musical scale.

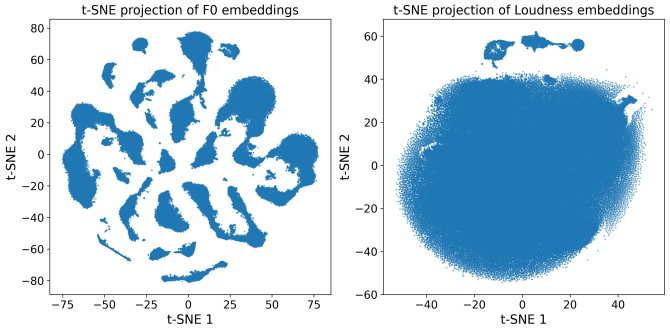

Fig. 4. t-SNE visualization of VAE embeddings for F_0 (left) and L (right)

To validate this observation, we analyzed the distribution of ground truth frequencies generated by CREPE, as shown in Figure 5. The frequency histogram reveals distinct peaks

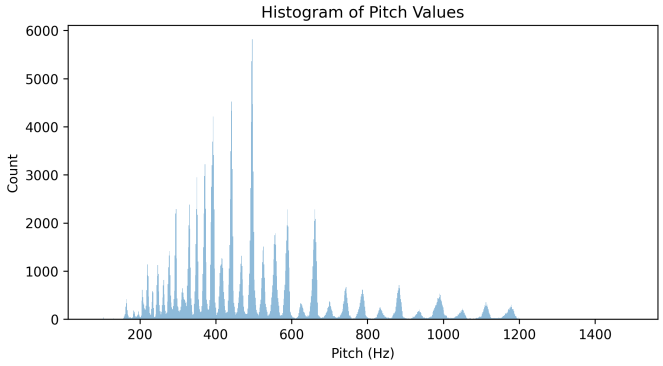

Fig. 5. Histogram of F_0 in the dataset extracted by CREPE model

corresponding to specific musical notes. The observed deviations around these peaks can be attributed to the estimation process of the CREPE model, which provides the ground truth annotations. These slight variations also capture natural phenomena in violin performance, such as vibrato and pitch transitions between notes.

An interpolation experiment was conducted to investigate the smoothness of the Audio Encoder's embedding space using linear interpolation: $z_{\text{interp}} = z_1 + \alpha(z_2 - z_1)$, where z_1 and z_2 are the embeddings of two different audio samples, and α varies from 0 to 1. Results are shown in Figure 6 with distinct behaviors for F_0 and L embeddings. Loudness interpolation demonstrates smooth transitions between samples, supported by t-SNE visualization showing a continuous distribution. However, F_0 interpolation exhibits binary switching behavior, particularly at $\alpha = 0.5$, due to the F_0 Decoder's output being a probability distribution over discrete frequencies and the inherent discreteness of musical pitch. Therefore t-SNE projections reinforce the characteristics of the embeddings F_0, which form distinct clusters reflecting discrete musical pitch, while loudness create a continuous distribution arising from smooth changes in amplitude variations.

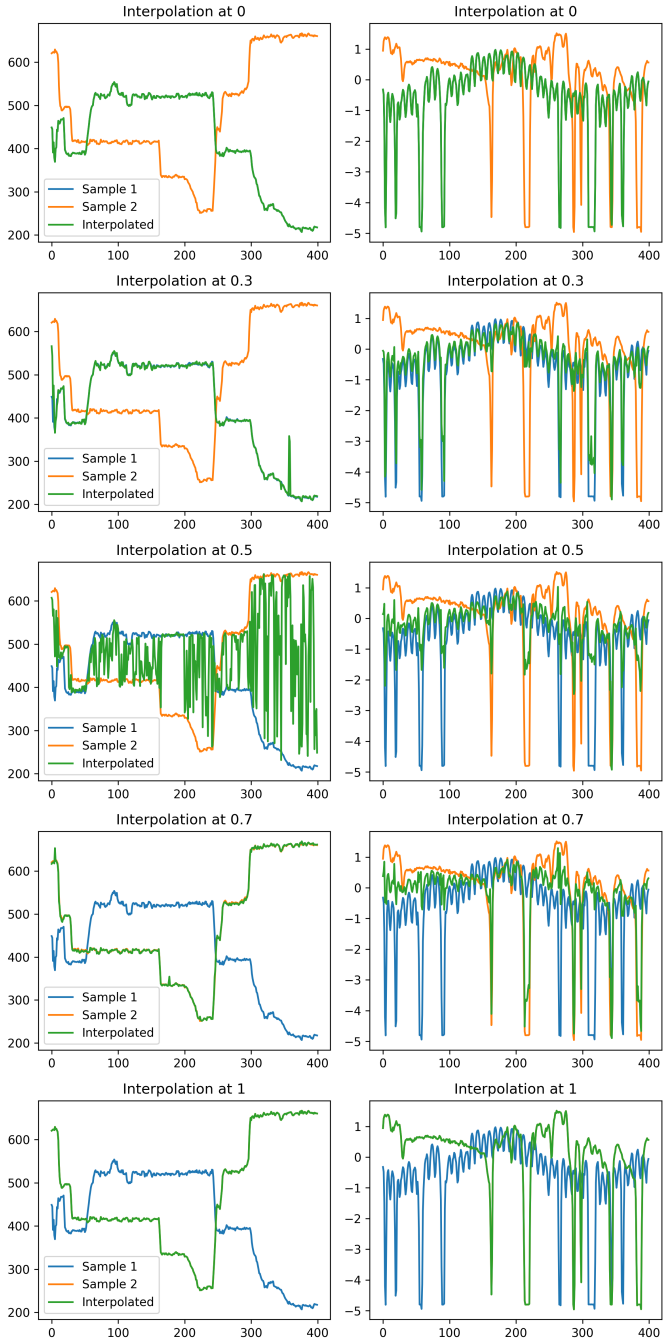

Fig. 6. Interpolation analysis of two audio samples showing F_0 (left) and L (right) characteristics at interpolation factors (α = 0, 0.3, 0.5, 0.7, 1).

C. DDSP Decoder

We trained four configurations of the DDSP Decoder on the MDB-synth-dataset using 30 minutes of audio for training and 20 minutes for validation. Each configuration underwent 200 epochs with 5000 randomly sampled clips per epoch. The training used a base learning rate of 2e-4 with learning rate decay on plateau. The validation results were generated using the remaining 20 minutes of audio.

Out of the four configurations i.e., GRU, Mamba, BiGRU

Model	MSS Loss (Mean)	MSS Loss (SD)
GRU (Baseline)	5.84 ± 0.02	0.59
Mamba	5.79 ± 0.02	0.58
BiGRU	5.63 ± 0.02	0.58
BiMamba	5.56 ± 0.02	0.57

and BiMamba, the BiMamba configuration resulted the lowest MSS loss of 5.56 ± 0.02, showing a significant improvement over the baseline GRU model's 5.84 ± 0.02. The bidirectional architectures (BiGRU and BiMamba) consistently outperformed their unidirectional counterparts. Additionally, the Mamba-based models demonstrated improved computational efficiency, reducing the training time from 58 seconds per 20K second audio epoch with GRU to 45 seconds with Mamba.

The spectrograms of the ground truth and synthesized audio using the BiMamba model are shown in Figure 7.

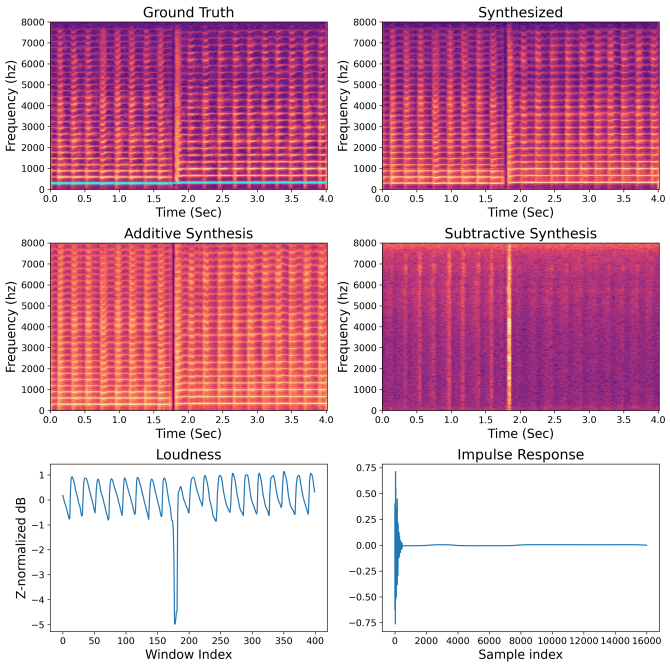

Fig. 7. Comparison of ground truth and synthesized audio spectrograms.

The synthesized spectrogram matches the ground truth in harmonic structure and temporal patterns. The fundamental frequency of the singal is given by the cyan line in the ground truth spectrogram. The additive synthesis captures the harmonic content while the subtractive synthesis shows noise components. The loudness curve represents amplitude variations and the impulse response shows acoustic modeling. The similarity between ground truth and synthesized spectrograms demonstrates the BiMamba model's effectiveness in violin audio synthesis.

Figure 8 shows the results of a timbre transfer task using the trained model on a singing voice track by Adele. The synthesized spectrograms exhibit characteristic violin harmonic structures and spectral envelopes while maintaining the temporal and pitch patterns of the original vocals.

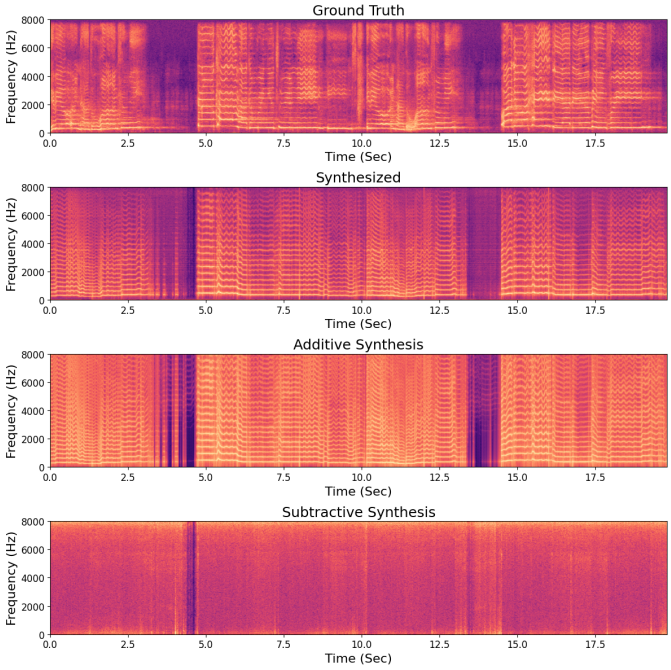

Fig. 8. Spectrograms of timbre transfer of a music sample.

V. Conclusion

A fully differentiable state-of-the-art modular framework using VAE based Audio Encoder combined with Mamba improved DDSP has been presented for smooth controlled audio generation. The Audio Encoder based latent space analysis with t-SNE projections reveled clustered F_0 embeddings and continuous L distributions. It shows that the model is capable for synthetic audio generation with musical note frequencies with smooth loudness interpolation capabilities. The DDSP Decoder, particularly in its bidirectional Mamba configuration (BiMamba), achieves superior performance with an MSS loss of 5.56 ± 0.02, improving upon the baseline GRU model's 5.84 ± 0.02. The Mamba-based architectures also demonstrate improved computational efficiency, reducing training time from 58 to 45 seconds per epoch. The system successfully preserves harmonic structures and temporal dynamics in violin synthesis. Furthermore, the model demonstrates effective timbre transfer capabilities, successfully mapping vocal characteristics to violin timbre while maintaining musical content. This study demonstrates an effective utilization of advantages of VAE's disentangled latent space with Mamba's efficient sequence modeling capabilities for high-quality audio synthesis and timbre transfer applications, while maintaining interpretability and control. Future work could explore extending this architecture to multi-instrumental synthesis and investigating more complex timbre transfer scenarios.

REFERENCES

[1] S. Shan, L. Hantrakul, J. Chen, M. Avent, and D. Trevelyan, "Differentiable wavetable synthesis," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4598–4602, 2022.

[2] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, "Style transfer of audio effects with differentiable signal processing," *Journal of the Audio Engineering Society*, vol. 70, pp. 708–721, september 2022.

[3] P.-A. Grumiaux and M. Lagrange, "Efficient bandwidth extension of musical signals using a differentiable harmonic plus noise model," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2023, p. 51, Dec 2023.

[4] N. Uzrad, O. Barkan, A. Elharar, S. Shvartzman, M. Laufer, L. Wolf, and N. Koenigstein, "Diffmoog: a differentiable modular synthesizer for sound matching," *CoRR*, vol. abs/2401.12570, 2024.

[5] J. Zhao, Y. Xu, X. Qian, H. Liu, M. D. Plumbley, and W. Wang, "Attention-based end-to-end differentiable particle filter for audio speaker tracking," *IEEE Open Journal of Signal Processing*, vol. 5, pp. 449–458, 2024.

[6] S. Clarke, N. Heravi, M. Rau, R. Gao, J. Wu, D. James, and J. Bohg, "Diffimpact: Differentiable rendering and identification of impact sounds," in *5th Annual Conference on Robot Learning*, 2021.

[7] Y. Liu, C. Jin, and D. Gunawan, "Ddsp-sfx: Acoustically-guided sound effects generation with differentiable digital signal processing," 2023.

[8] G. Richard, P. Chouteau, and B. Torres, "A fully differentiable model for unsupervised singing voice separation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Seoul, South Korea), Apr. 2024.

[9] K. Schulze-Forster, G. Richard, L. Kelley, C. S. J. Doire, and R. Badeau, "Unsupervised music source separation using differentiable parametric source models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1276–1289, 2023.

[10] M. H. Erol, A. Senocak, J. Feng, and J. S. Chung, "Audio mamba: Bidirectional state space model for audio representation learning," 2024.

[11] D. Lee and J.-W. Choi, "Deft-mamba: Universal multichannel sound separation and polyphonic audio classification," 2024.

[12] K. Miyazaki, Y. Masuyama, and M. Murata, "Exploring the capability of mamba in speech applications," 2024.

[13] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," 2019.

[14] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," 2021.