Modulation Discovery with Differentiable Digital Signal Processing

Christopher Mitcheltree$^\flat$, Hao Hao Tan$^\sharp$, Joshua D. Reiss$^\flat$

$^\flat$Centre for Digital Music, Queen Mary University of London, UK
$^\sharp$Independent Researcher, Singapore

Abstract—**Modulations are a critical part of sound design and music production, enabling the creation of complex and evolving audio. Modern synthesizers provide envelopes, low frequency oscillators (LFOs), and more parameter automation tools that allow users to modulate the output with ease. However, determining the modulation signals used to create a sound is difficult, and existing sound-matching / parameter estimation systems are often uninterpretable black boxes or predict high-dimensional framewise parameter values without considering the shape, structure, and routing of the underlying modulation curves. We propose a neural sound-matching approach that leverages modulation extraction, constrained control signal parameterizations, and differentiable digital signal processing (DDSP) to discover the modulations present in a sound. We demonstrate the effectiveness of our approach on highly modulated synthetic and real audio samples, its applicability to different DDSP synth architectures, and investigate the trade-off it incurs between interpretability and sound-matching accuracy. We make our code and audio samples available and provide the trained DDSP synths in a VST plugin.**

1. INTRODUCTION

A key aspect of audio production is introducing dynamic variation and movement. This is achieved through *modulation* (mod.), which uses one signal to control a parameter of another signal or component within a synthesizer, allowing the sound to evolve over time. In modern synthesizers, this may involve an envelope generator modulating amplitude to shape a sound's attack and decay, or a low-frequency oscillator (LFO) modulating parameters such as pitch (vibrato), filter cutoff (wah-wah), or amplitude (tremolo). Modulation is ubiquitous in contemporary sound design and is especially crucial in electronic music, where artists continually push sonic boundaries. For reference, $\sim 98\%$ of the presets in the default library of `Serum`[1], a popular commercial synthesizer, use modulation. Modern soft synths like `Vital`[2] and `Serum` emphasize modulation as a core feature, offering drawable XY modulation grids and advanced modulation routing options such as wavetable position and spectral warping.

Parameter estimation for sound synthesizers, also known as *sound matching*, is an active area of research focused on predicting synth parameters that best reproduce a target sound [1]–[4]. Some approaches are broadly applicable across all types of synthesizers [5]–[9], while others focus on specific techniques such as additive [10], [11], wavetable [12]–[14], frequency modulation [15], [16], waveshaping [17], and modular synthesis [18], [19]. Differentiable digital signal processing (DDSP) based methods formulate the parameter estimation problem as a self-supervised learning task, wherein a differentiable synthesizer is optimized to re-construct a target audio signal, and this process implicitly infers the underlying synth parameters. This class of methods shares similar objectives with the field of neural audio synthesis [20].

However, current research does not address the extraction and discovery of *interpretable* modulation signals. Most works produce static, global synth parameters with no variation across time, or infer
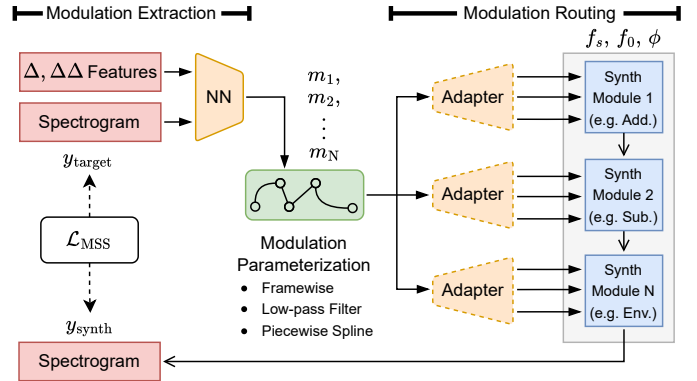
[1] https://xferrecords.com/products/serum-2
[2] https://vital.audio/

Fig. 1: Overview of the modulation discovery process through modulation extraction, parameterization, and routing using a DDSP synth. Orange blocks are neural networks, dashed blocks are optional, and blue blocks are differentiable and may contain learnable weights for sound matching.

Fig. 2: A 2D drawable modulation grid in the `Vital` soft synth.

framewise mod. signals that are high-dimensional, difficult to interpret, or overly complex, which are in contrast to the intuitive modulation curves crafted by human sound designers. While parallel work exists on extracting time-varying control signals for audio effects [21], [22], there has been limited exploration of this approach in the area of sound synthesis. Motivated by this research gap, and inspired by related work on audio effect discovery [23] and differentiable curve parameterizations [24]–[26], we propose a technique for discovering interpretable modulation signals present in a sound.

Our contributions are as follows:

- We introduce a modulation discovery approach (see Figure 1) that leverages modulation extraction, DDSP synthesis, and self-supervised sound matching to implicitly infer underlying modulations in arbitrary target audio.
- We define constrained, differentiable modulation signal parameterizations based on low-pass filtering and splines that ensure interpretable curves resembling those made by sound designers.
- We demonstrate the effectiveness of our approach on reconstructing highly modulated audio samples, both synthetic and real, and investigate the trade-off between mod. signal interpretability and sound matching accuracy for different DDSP synth architectures.
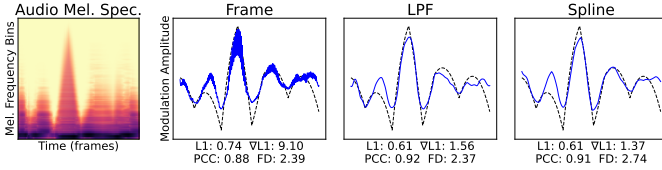
Fig. 3: Modulation extraction with the framewise, low-pass filtered, and piecewise Bézier curve parameterizations. The ground truth signal is dashed and the four mod. signal distance measures from Section 3.1 are computed.

2. METHODOLOGY

We propose a modulation discovery approach (shown in Figure 1) consisting of three steps: modulation routing, extraction, and parameterization, that allows interpretable mod. signals to be uncovered in black / gray-box time-varying audio sources. While this work focuses on modulations of software synths for electronic music, our approach is general and can be applied to other instruments and audio domains.

2.1. Modulation Routing with DDSP

First, a DDSP synth architecture is selected that consists of controllable, time-varying synth modules and reflects the routing and high-level types of modulations to be discovered. In this work, we focus on additive (Add.), subtractive (Sub.), and envelope (Env.) modulations. Leveraging DDSP provides interpretability, computational efficiency, and strong guardrails on generated audio. Inspired by the popularity and expressiveness of soft synths like `Serum` and `Vital`, we build a synth (called Mod. Synth) with three differentiable modules in series: a wavetable oscillator with controllable wavetable position, a resonant filter with controllable coefficients, and an envelope. We implement the wavetable module using `PyTorch`'s `grid_sample` function and use the time-varying biquad filter from [22] for the filter module.

2.2. Modulation Extraction through Sound Matching

Next, a neural network extracts low-dimensional (in this work 1D) mod. signals for each synth module via a sound matching setup. We use LFO-net from [21] which has a high inductive bias for extracting modulations and can be trained with limited data. A simple multilayer perceptron adapter converts these mod. signals to higher dimensions for synth modules that require it (e.g. filter coefficients). Training is self-supervised where input and output audio is compared using a perceptual loss function. This information bottleneck + adapter technique lowers sound matching accuracy, but provides interpretability and causal controllability of the synth during inference. Leveraging sound matching to discover modulations means this approach can be applied to any evolving audio and has the flexibility to uncover modulation shapes and combinations different from those present in the target audio.

2.3. Modulation Parameterization

Finally, we propose three differentiable parameterizations for the extracted low-dimensional mod. signals and investigate their curve fitting and sound matching abilities in the experiments section. Figure 3 provides a visualization of them and modulation extraction.

Framewise (Frame) — By default, mod. signals are extracted frame-by-frame at a control rate f_{MS} much slower than the sample rate (i.e. $f_{MS} \approx \frac{f_s}{100}$). Most DDSP systems use this parameterization.

Low-pass Filter (LPF) — We post-process extracted mod. signals at f_{MS} with a low-pass filter and cutoff frequency $f_c < 20$ Hz, which ensures they are smooth and similar to LFO signals.

2D Bézier Curve (Spline) — Inspired by the drawable XY modulation grids of soft synths (shown in Figure 2), we implement a piecewise 2D Bézier curve [27] mod. signal parameterization $B(t)$.

We define $K \in \mathbb{Z}_{>0}$ segments on intervals u_k with each segment B_k consisting of $n \in \mathbb{Z}_{>0}$ control points $P_{k,i}$ such that:

$$P_{k,i} = (x_{k,i}, y_{k,i}) \in \mathbb{R}^2, \quad i = 0, 1, \ldots, n \quad (1)$$

$$B_k(u_k) = \sum_{i=0}^{n} \binom{n}{i} (1 - u_k)^{n-i} u_k^i P_{k,i} \quad (2)$$

where $k = \min(\lfloor K \cdot t \rfloor, K - 1), \quad u_k = K \cdot t - k \quad (3)$

$$B(t) = B_k(u_k), \quad t \in [0, 1] \quad (4)$$

To ensure C^0 (positional) continuity and covering all of t, we enforce

$$P_{k,n} = P_{k+1,0}, \quad \forall k = 0, 1, \ldots, K - 2 \quad (5)$$

$$0 = x_{0,0} < x_{0,1} < \cdots < x_{0,n} = x_{1,0} < \cdots < x_{K-1,n} = 1 \quad (6)$$

Bézier curves offer several advantages, including compact, low-dimensional representations of smooth signals, differentiability and efficient computation due to their closed-form expression, and interpretability with bounded outputs, as the curve remains entirely within the convex hull of its control points.

3. EXPERIMENTS

We conduct three experiments to evaluate our proposed approach: its *modulation extraction* capabilities on unseen real-world mod. signals (Sec. 3.1), and its *modulation discovery* ability for synthetic (Sec. 3.2) and real-world audio (Sec. 3.3). We provide audio samples, VST plugins, code, and additional details at the accompanying website.[3]

Training — All experiments train LFO-nets with 1.0 ± 0.2M parameters and a temporal receptive field of 1471 frames for 30 epochs on a 60-20-20% train-val-test split, with a batch size of 32, the Schedule-Free AdamW optimizer [28], a learning rate of 0.005 for models / 0.05 for learnable synth modules, and using multi-resolution STFT loss (MSS) [29]. Spline mod. signals are linearly interpolated over the first $\frac{2}{3}$ of training from linear to cubic which encourages learning high-level structure before becoming detailed and curvy. In Experiments 3.2 and 3.3 for all parameterizations, Gaussian noise with $\sigma = 0.33$ is added (and linearly decayed over the first $\frac{2}{3}$ of training) to the predicted wavetable position of Mod. Synth to encourage the use of all available learnable wavetable positions.

Input Features — Input to all models is a 3-channel Mel spectrogram (128 Mel bins, 1024 FFT size, 96 sample hop length) with 50 ms delta and delta-delta features [30] computed from three seconds of $f_s = 48$ kHz audio which results in 1501 frames and a f_{MS} of 500 Hz. Since pitch modulations are not being investigated, the f_0 of the input audio is provided to synths and phase (ϕ) is randomized.

3.1. Modulation Extraction (Real-world Data)

The first experiment evaluates how well LFO-net and the three parameterization methods can extract mod. signals when a white-box synth architecture is used. Audio is generated and reconstructed with a frozen Mod. Synth (see Section 2.1), which means extracted mod. signals can be compared directly to their ground truth counterparts.

Parameterization — We use a cubic 2D Bézier curve with 24 segments for the spline method, so it can approximate up to an ~ 8 Hz sinusoidal mod. signal. For a fair comparison, we use a Blackman windowed-sinc low-pass filter with an $f_c = 8$ Hz cutoff frequency and $\left\lceil \frac{f_{MS}}{8} \right\rceil$ taps for the LPF method.

DDSP Synth — The Mod. Synth wavetable uses a LPF for anti-aliasing with $f_c = 0.9 \cdot \frac{f_s}{2}$. The filter module is a low-pass biquad filter with 100 Hz $\leq f_c \leq 8$ kHz and $\frac{\sqrt{2}}{2} \leq Q \leq 4$.

---

[3] https://christhetree.github.io/mod_discovery/

Table 1: Modulation extraction evaluation results on unseen real-world data.

Mod.	Method	$L_1 \downarrow$	$\nabla L_1 \downarrow$	PCC $\uparrow$	FD $\downarrow$
Add.	Frame	**2.41±0.08**	2.51±0.06	**0.46±0.03**	5.36±0.09
	LPF	2.42±0.07	1.78±0.03	0.46±0.02	**5.29±0.07**
	Spline	2.44±0.09	**1.71±0.03**	0.45±0.03	5.33±0.09
Sub.	Frame	1.45±0.01	2.04±0.03	**0.76±0.01**	4.33±0.05
	LPF	**1.44±0.01**	1.52±0.02	**0.76±0.01**	4.28±0.03
	Spline	1.45±0.01	**1.48±0.02**	0.75±0.01	**4.27±0.04**
Env.	Frame	**2.11±0.02**	3.51±0.09	**0.68±0.01**	5.25±0.06
	LPF	2.12±0.02	1.85±0.02	**0.68±0.01**	5.22±0.06
	Spline	**2.11±0.01**	**1.76±0.02**	**0.68±0.01**	**5.18±0.06**
All	Rand. Spline	3.35±0.01	8.96±0.04	0.00±0.01	7.10±0.05
	Rand. Frame	3.69±0.01	166.80±0.13	0.00	7.77±0.05

Table 2: Extracted, target, and random baseline modulation signal metrics. Values closer to their corresponding target values are better.

Method	Total Variation	Turning Points	Spectral Entropy
Frame	4.40±0.09	157.01±2.79	0.490±0.002
LPF	2.28±0.01	14.46±0.20	**0.453±0.002**
Spline	**2.10±0.02**	**10.05±0.12**	0.451±0.002
Target	1.69±0.01	1.21±0.06	0.464±0.006
Rand. Spline	9.50±0.03	26.88±0.08	0.560±0.001
Rand. Frame	333.61±0.25	666.79±0.36	0.935

Table 3: Modulation discovery evaluation results for synthetic data.

Mod.	Proc.	Method	$L_1 \downarrow$	$\nabla L_1 \downarrow$	PCC $\uparrow$	FD $\downarrow$
Add.	LLS 1	Frame	1.11±0.02	2.16±0.12	0.47±0.03	3.36±0.04
		LPF	**1.08±0.03**	1.42±0.03	**0.50±0.02**	**3.19±0.06**
		Spline	1.16±0.01	**1.37±0.03**	0.43±0.02	3.27±0.03
	LLS 3	Frame	0.91±0.02	3.08±0.13	0.67±0.01	3.38±0.13
		LPF	**0.88±0.02**	1.58±0.04	**0.69±0.01**	2.90±0.08
		Spline	0.94±0.01	**1.53±0.03**	0.66±0.01	**2.87±0.03**
Sub.	LLS 1	Frame	0.62±0.03	2.90±0.25	**0.82±0.01**	2.84±0.18
		LPF	**0.61±0.03**	**1.03±0.03**	0.82±0.02	2.27±0.12
		Spline	0.67±0.04	1.04±0.06	0.79±0.03	**2.24±0.12**
	LLS 3	Frame	0.50±0.01	2.66±0.14	0.89±0.01	2.69±0.17
		LPF	**0.48±0.01**	1.06±0.02	**0.90±0.01**	1.97±0.05
		Spline	0.52±0.01	**1.04±0.02**	0.88±0.01	**1.88±0.03**
Env.	LLS 1	Frame	1.28±0.02	2.16±0.07	0.35±0.01	3.90±0.13
		LPF	1.27±0.02	1.43±0.04	0.37±0.01	3.61±0.07
		Spline	**1.25±0.02**	**1.39±0.03**	**0.38±0.01**	**3.54±0.06**
	LLS 3	Frame	0.96±0.02	3.74±0.14	0.67±0.01	3.62±0.18
		LPF	**0.92±0.02**	1.68±0.04	**0.69±0.01**	3.00±0.09
		Spline	0.95±0.02	**1.60±0.03**	0.68±0.01	**2.91±0.06**
All	LLS 1	Rand. S.	1.38±0.02	1.91±0.04	0.15±0.01	3.65±0.04
	LLS 3	Rand. S.	1.33±0.02	2.83±0.04	0.30±0.01	3.47±0.05

Datasets — Models are trained on 1024 datapoints of three randomly generated spline mod. signals (one for each synth module) using a 1-3 degree piecewise 1D Bézier curve consisting of 1-8 segments. Segment intervals are random with a min. length of half a uniform segment length. Filter resonance values (Q) are constant and sampled log-uniformly. 10 wavetables from `Ableton Live 12`'s stock library are used for initializing Mod. Synth's additive module. They consist of 4 to 256 positions, 1024 samples, and are chosen to be tonal. 10 models are trained for each wavetable using different random seeds.

All 100 models are tested on an unseen, real-world mod. signal dataset consisting of modulation curves from `Vital`'s default preset library. Signals that consist of > 24 segments, span $< 50\%$ of the modulation range, or are $> 50\%$ flat are removed, leaving 52 unique curves. 1024 randomly sampled datapoints from the resulting ~ 132k unique combinations of three mod. signals are used for evaluation.

Evaluation — We compare Add, Sub, and Env. mod. signals using four different measures: L_1 distance, first-order central difference L_1 distance (∇L_1), Pearson correlation coefficient (PCC), and Fréchet distance (FD). They are chosen to cover shift, scale, and time-warping invariance to compare shape, structure, absolute, and relative differences between two modulations. We also calculate normalized total variation (TV), turning points (TP), and spectral entropy (SE) for each signal. Uniformly sampled random spline and framewise baselines are included as an anchor to compare against.

Results — From Tables 1 and 2, we observe that all parameterizations minimize absolute differences (L_1) between mod. signals. Unsurprisingly, Spline and LPF better capture smooth variations (∇L_1), with Spline best matching the target TV and TP metrics. LFO-net also excels at extracting envelope and filter modulations over wavetable, likely due to their simpler, more visible effect on the input features.

3.2. Modulation Discovery (Synthetic Data)

The second experiment evaluates how well our modulation routing and DDSP sound matching approach can discover modulations for a gray-box synth. This experiment is identical to Section 3.1, except only synthetic mod. signals are used and now Mod. Synth has a learnable, 16-position, 1024-sample wavetable initialized with Gaussian noise ($\sigma = 0.01$) and the learnable time-varying biquad filter from [22].

Evaluation — Discovered mod. signals for learnable synth modules can no longer be directly compared to their respective ground truth signals since they may be shifted, flipped, and scaled depending on the stochastic gradient descent optimization process. For example, if the learned wavetable module happens to be in reverse order of the original in the dataset, then the discovered additive modulations will be mirrored vertically. Therefore, we post-process extracted mod. signals with a linear least squares (LLS) solver using their corresponding ground truth signal as the target (labeled LLS 1). Discovered mod. signals may also combine together and fulfill different purposes to match the target sound. For example, some of the envelope modulations in the target audio may be achieved using a mixture of wavetable and filter modulations in the learned synth. Consequently, we use the LLS solver to make a linear combination of all three discovered mod. signals for each ground truth signal (labeled LLS 3). We then apply the same mod. signal distance metrics as before.

For measuring sound matching quality, we use MSS, Mel-frequency cepstral coefficients (MFCC) L_1 distance, and two Fréchet Audio Distances (FAD) [31], [32] which have been shown to correlate with human perception [33]. We include the random spline baseline from before (mod. signals are only randomized during testing, not training) as well as an Oracle baseline that always uses the ground truth mod. signals and serves as a consistent point of comparison for evaluating the sound matching ability of the LFO-net + Mod. Synth system.

Results — Table 3 shows that LLS 3 significantly outperforms LLS 1 across all signals and parameterizations, suggesting that the modulation discovery process distributes modulations across different synth modules. This is particularly evident for Sub. and Env., which make similar perceptual changes to the output audio. LPF and Spline again best match the ground-truth signals, benefiting from strong inductive biases. Compared to the previous experiment, extracted modulation evaluation distances are more consistent and uniformly distributed, highlighting the advantage of a flexible sound matching approach with learnable synth modules. Looking at the audio similarity metrics in Table 4, all methods sometimes surpass the Oracle baseline on MSS and FAD. This indicates that discovered modulations can better optimize the perceptual MSS loss and that MFCC is the most reliable metric for evaluating underlying modulation structure in audio.

Table 4: Sound matching audio similarity metrics for synthetic data.

Method	MSS ↓	MFCC ↓	FAD MS-CLAP ↓	FAD EnCodec ↓
Frame	**0.94±0.01**	1.60±0.02	**109.83± 4.49**	12.92± 1.78
LPF	0.95±0.02	**1.54±0.03**	115.70± 7.13	**12.73± 1.58**
Spline	1.02±0.02	1.76±0.03	131.93± 6.26	16.27± 1.77
Oracle	1.05±0.03	1.48±0.06	111.43± 8.83	16.80± 1.12
Rand. S.	10.51±6.25	4.67±0.09	764.73±55.00	181.60±85.49

3.3. Modulation Discovery (Real-world Audio)

The last experiment evaluates how well our modulation discovery approach generalizes to real-world audio, black-box synths, and different DDSP synth architectures and their modulation routings.

Dataset — This experiment is identical to Section 3.2, except the dataset now consists of complex, highly modulated audio from the popular soft synth Serum. All preset patches in the "Bass (Hard)" category with both wavetable and filter modulations are selected and notes A0 to C8 are rendered, resulting in 44 minutes of audio and 880 datapoints total. Three different DDSP synths and their models are trained 20 times on the dataset with unique random seeds.

Evaluation — We test Mod. Synth, a variant of Mod. Synth with Shan et al.'s [13] wavetable module which uses time-varying attention across wavetable positions, and Engel et al.'s [10] harmonic oscillator + filtered noise synth (100 harmonics, 65 noise bands) with a learnable envelope at its output. Since the underlying DSP mod. signals of the target audio are now unknown, we compute 1D audio features (root mean square loudness (RMS) and spectral flatness (SF)) at a frame rate of f_{MS} to use as approximations and apply the LLS 3 post-processing step to extracted mod. signals before comparing. We hypothesize that RMS should correlate with sub. and env. modulations, while SF is perceived as tonality and should correlate with the add. and sub. modulations. To determine the trade-off between sound matching quality and interpretability incurred by extracting low-dimensional mod. signals for each synth module (see Section 2.2), we include a granular mod. signal baseline (Gran.) that uses a high-dimensional framewise mod. signal for its synth modules. This eliminates the information bottleneck and is the default setup in most DDSP systems. Figure 4 visualizes some of the discovered modulations.

Results — Overall, sound matching and modulation extraction performance declines, reflecting the complexity of the Serum audio, the limited expressiveness of the selected synths, and the approximation of the "ground-truth" signals. Table 5 shows LPF consistently yields the closest match to RMS and SF modulations across architectures, with Spline usually a close second. Additional experiments with spectral centroid and bandwidth also give similar results. Figure 4 illustrates how the Frame method overfits each timestep to maximize sound matching, a trend echoed in Table 6: parameterizations with more degrees of freedom better optimize perceptual loss, but sacrifice interpretability. Importantly, the drop in matching accuracy from Gran. to LPF is far smaller for Mod. Synth and Shan et al. than for Engel et al., underscoring the advantage of choosing a DDSP architecture with lower-dimensional control parameters for its modules.

4. CONCLUSION

We introduce a framework that uncovers modulations present in audio through extraction, parameterization, and DDSP synth sound matching. We demonstrate its ability to discover unseen and approximate ground truth mod. signals in synthetic and real-world audio. Our results show that a low-pass filter parameterization best balances expressive sound matching with human readability, while a spline parameterization maximizes interpretability. We believe that creative applications with unconventional input audio and analyzing the symmetry of discovered modulations are compelling directions for future work.
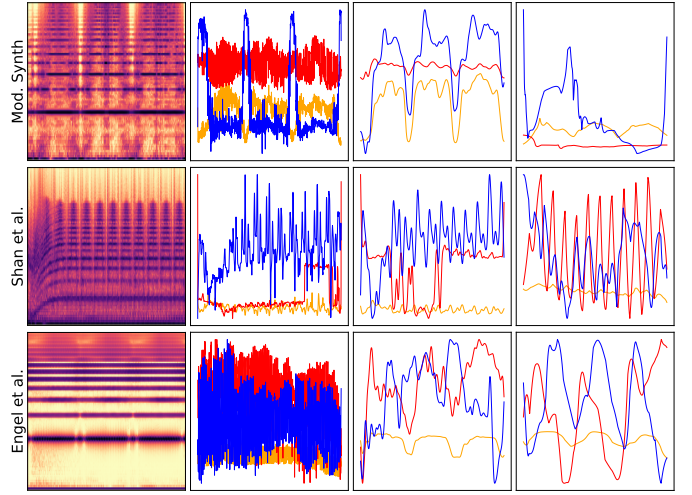
Fig. 4: Discovered additive (red), subtractive (blue), and envelope (orange) modulations in Serum test dataset audio samples using different DDSP synths and Frame (left), LPF (center), and Spline (right) mod. signal parameterizations.

Table 5: Modulation discovery evaluation results for real-world data.

Arch.	Mod.	Method	L_1 ↓	∇L_1 ↓	PCC ↑	FD ↓
Mod. Synth	RMS	Frame	1.82±0.01	4.03±0.24	0.42±0.01	**6.94±0.10**
		LPF	**1.72±0.02**	**2.91±0.01**	**0.48±0.01**	6.97±0.10
		Spline	1.80±0.03	2.92±0.01	0.44±0.01	7.24±0.08
	SF	Frame	2.67±0.06	7.85±0.38	**0.54±0.01**	**3.66±0.04**
		LPF	**2.63±0.04**	6.15±0.01	**0.54±0.01**	3.75±0.03
		Spline	2.97±0.08	6.20±0.01	0.46±0.01	3.92±0.03
Shan et al.	RMS	Frame	1.85±0.02	4.06±0.18	0.41±0.01	**6.68±0.11**
		LPF	**1.75±0.02**	**2.91±0.01**	**0.45±0.01**	6.92±0.07
		Spline	1.81±0.03	2.93±0.01	0.43±0.01	7.19±0.06
	SF	Frame	2.70±0.03	7.86±0.32	**0.53±0.01**	**3.66±0.04**
		LPF	**2.63±0.04**	6.15±0.01	0.52±0.01	3.76±0.03
		Spline	3.02±0.07	6.21±0.01	0.45±0.01	3.95±0.02
Engel et al.	RMS	Frame	2.18±0.02	4.31±0.14	0.22±0.01	7.54±0.09
		LPF	**1.80±0.02**	**2.95±0.01**	**0.46±0.01**	**7.24±0.08**
		Spline	1.86±0.02	2.96±0.01	0.43±0.01	7.36±0.07
	SF	Frame	3.65±0.06	9.32±0.45	0.25±0.02	3.95±0.03
		LPF	**2.89±0.04**	5.89±0.01	**0.43±0.01**	**3.87±0.02**
		Spline	2.98±0.05	**5.88±0.01**	0.39±0.01	3.92±0.02
All	RMS	Rand. S.	2.20±0.01	2.96±0.01	0.22±0.01	7.82±0.02
	SF	Rand. S.	3.92±0.01	6.28±0.01	0.19±0.01	4.13±0.01

Table 6: Sound matching audio similarity metrics for real-world data.

Arch.	Method	MSS ↓	MFCC ↓	FAD ↓ MS-CLAP	FAD ↓ EnCodec
Mod. Synth	Gran.	**1.47±0.01**	**2.31±0.02**	**311.24± 4.91**	**29.21± 0.95**
	Frame	1.60±0.05	2.56±0.10	361.52± 9.51	34.96± 2.64
	LPF	1.63±0.01	2.48±0.02	395.32± 9.72	37.81± 0.89
	Spline	1.67±0.01	2.58±0.04	411.73±17.39	40.19± 1.52
	Rand. S.	2.79±0.11	3.40±0.07	505.15±50.36	53.50±12.49
Shan et al.	Gran.	**1.48±0.01**	**2.28±0.03**	**306.53± 7.11**	**28.48± 1.23**
	Frame	1.61±0.02	2.54±0.04	381.84±17.17	37.91± 2.08
	LPF	1.66±0.02	2.60±0.07	434.43±25.58	41.37± 1.77
	Spline	1.69±0.01	2.60±0.05	469.68±15.07	44.19± 1.12
	Rand. S.	3.06±0.24	3.53±0.11	509.76±43.82	83.02±17.31
Engel et al.	Gran.	**1.81±0.04**	**2.99±0.05**	**461.24±15.16**	**35.04± 1.11**
	Frame	2.06±0.05	3.27±0.04	667.88±27.87	57.69± 1.94
	LPF	3.18±0.01	5.94±0.01	699.92± 9.75	91.29± 0.89
	Spline	3.19±0.01	5.94±0.01	728.35±10.65	93.23± 0.75
	Rand. S.	4.42±0.15	6.28±0.14	644.06±21.19	99.41± 1.81

REFERENCES

[1] S. Heise, M. Hlatky, and J. Loviscach, "Automatic cloning of recorded sounds by software synthesizers," in *127th Convention of the Audio Engineering Society*, Oct. 2009.

[2] K. Tatar, M. Macret, and P. Pasquier, "Automatic synthesizer preset generation with PresetGen," *Journal of New Music Research*, vol. 45, no. 2, pp. 124–144, 2016.

[3] J. Shier, "The synthesizer programming problem: improving the usability of sound synthesizers," Dec. 2021, University of Victoria Master's Thesis, available at https://dspace.library.uvic.ca/handle/1828/13593.

[4] C. J. Steinmetz, C. Uhle, F. Everardo, C. Mitcheltree, J. K. McElveen, J.-M. Jot, and G. Wichern, "Audio signal processing in the artificial intelligence era: Challenges and directions," *Journal of the Audio Engineering Society*, vol. 73, pp. 406–428, July 2025, https://doi.org/10.17743/jaes.2022.0209.

[5] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic programming of VST sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 150–159, Apr. 2018, https://doi.org/10.1109/TETCI.2017.2783885.

[6] 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, p. 302, 2019.

[7] Y. Yang, Z. Jin, C. Barnes, and A. Finkelstein, "White box search over audio synthesizer parameters," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[8] F. Bruford, F. Blang, and S. Nercessian, "Synthesizer sound matching using audio spectrogram transformers," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2024.

[9] B. Hayes, C. Saitis, and G. Fazekas, "Audio synthesizer inversion in symmetric parameter spaces with approximately equivariant flow matching," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2025, https://doi.org/10.48550/arXiv.2506.07199.

[10] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Apr. 2020.

[11] N. Masuda and D. Saito, "Improving semi-supervised differentiable synthesizer sound matching for practical applications," *IEEE Transactions on Speech and Audio Processing*, vol. 31, pp. 863–875, Jan. 2023, https://doi.org/10.1109/TASLP.2023.3237161.

[12] C. Mitcheltree and H. Koike, "SerumRNN: Step by step audio VST effect programming," in *Proceedings of the International Conference on Artificial Intelligence in Music, Sound, Art, and Design (EvoMUSART)*, Apr. 2021, pp. 218–234, https://doi.org/10.48550/arXiv.2104.03876.

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

[14] C.-Y. Yu and G. Fazekas, "GOLF: A singing voice synthesiser with glottal flow wavetables and LPC filters," *Transactions of the International Society for Music Information Retrieval (ISMIR)*, Dec 2024.

[15] F. Caspe, A. McPherson, and M. Sandler, "DDX7: Differentiable FM synthesis of musical instrument sounds," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[16] Z. Chen, Y. Jing, S. Yuan, Y. Xu, J. Wu, and H. Zhao, "Sound2Synth: Interpreting sound via FM synthesizer parameters estimation," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 6, Jul. 2022, pp. 4921–4928, https://doi.org/10.24963/ijcai.2022/682.

[17] B. Hayes, C. Saitis, and G. Fazekas, "Neural waveshaping synthesis," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[18] J. Turian, J. Shier, G. Tzanetakis, K. McNally, and M. Henry, "One billion audio sounds from GPU-enabled modular synthesis," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Sep. 2021, pp. 222–229, https://doi.org/10.23919/DAFx51585.2021.9768246.

[19] N. Uzrad, O. Barkan, A. Elharar, S. Shvartzman, M. Laufer, L. Wolf, and N. Koenigstein, "DiffMoog: a differentiable modular synthesizer for sound matching," 2024, arXiv preprint: https://arxiv.org/abs/2401.12570.

[20] B. Hayes, J. Shier, G. Fazekas, A. McPherson, and C. Saitis, "A review of differentiable digital signal processing for music and speech synthesis," *Frontiers in Signal Processing*, vol. 3, p. 1284100, Jan. 2024, https://doi.org/10.3389/frsip.2023.1284100.

[21] C. Mitcheltree, C. J. Steinmetz, M. Comunità, and J. D. Reiss, "Modulation extraction for LFO-driven audio effects," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2023.

[22] C.-Y. Yu, C. Mitcheltree, A. Carson, S. Bilbao, J. D. Reiss, and G. Fazekas, "Differentiable all-pole filters for time-varying audio systems," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2024.

[23] C. J. Steinmetz and J. D. Reiss, "Steerable discovery of neural audio effects," *Neural Information Processing Systems Workshop on Machine Learning for Creativity and Design*, Dec. 2021, https://arxiv.org/abs/2112.02926.

[24] D. Ha and D. Eck, "A neural representation of sketch drawings," in *Proceedings of the International Conference on Learning Representations (ICLR)*, https://openreview.net/forum?id=Hy6GHpkCW.

[25] R. Nakano, "Neural painters: A learned differentiable constraint for generating brushstroke paintings," in *Neural Information Processing Systems Workshop on Machine Learning for Creativity and Design*, https://arxiv.org/abs/1904.08410.

[26] E. Scharwächter, J. Lennartz, and E. Müller, "Differentiable segmentation of sequences," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021, https://openreview.net/forum?id=4T489T4yav.

[27] P. Bézier, "Procédé de définition numérique des courbes et surfaces non mathématiques," *Automatisme*, vol. 13, no. 5, pp. 189–196, 1968.

[28] A. Defazio, X. A. Yang, H. Mehta, K. Mishchenko, A. Khaled, and A. Cutkosky, "The road less scheduled," 2024, arXiv preprint: https://arxiv.org/abs/2405.15682.

[29] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 6199–6203.

[30] S. Furui, "Speaker-independent isolated word recognition based on emphasized spectral dynamics," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 11, 1986, pp. 1991–1994.

[31] B. Elizalde, S. Deshmukh, and H. Wang, "Natural language supervision for general-purpose audio representations," 2023, arXiv preprint: https://arxiv.org/abs/2309.05767.

[32] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *Transactions on Machine Learning Research*, 2023, https://openreview.net/forum?id=ivCd8z8zR2.

[33] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Proceedings of the International Speech Communication Association Conference (Interspeech)*, 2019, pp. 2350–2354.