PREPRINT Accepted at the 28th International Conference on Digital Audio Effects (DAFx25), Ancona, Italy, 2 - 5 September 2025

MORPHDRIVE: LATENT CONDITIONING FOR CROSS-CIRCUIT EFFECT MODELING AND A PARAMETRIC AUDIO DATASET OF ANALOG OVERDRIVE PEDALS

Francesco Ardan Dal Rí, Domenico Stefani, Luca Turchet and Nicola Conci

DISI - Department of Information Engineering and Computer Science
University of Trento
Trento, IT
francesco.dalri-2@unitn.it, domenico.stefani@unitn.it

ABSTRACT

In this paper, we present an approach to the neural modeling of overdrive guitar pedals with conditioning from a cross-circuit and cross-setting latent space. The resulting network models the behavior of multiple overdrive pedals across different settings, offering continuous morphing between real configurations and hybrid behaviors. Compact conditioning spaces are obtained through unsupervised training of a variational autoencoder with adversarial training, resulting in accurate reconstruction performance across different sets of pedals. We then compare three Hyper-Recurrent architectures for processing, including dynamic and static Hyper-RNNs, and a smaller model for real-time processing. Additionally, we present *pOD-set*, a new open dataset including recordings of 27 analog overdrive pedals, each with 36 gain and tone parameter combinations totaling over 97 hours of recordings. Precise parameter setting was achieved through a custom-deployed recording robot.

1. INTRODUCTION

Virtual analog modeling (VA) of audio effects has long been a topic of interest in audio signal processing [1, 2, 3]. VA methods aim to emulate the behavior of analog circuits in the digital domain to replicate the sonic characteristics of vintage and modern hardware. Among these, overdrive pedals have been a popular target for VA modeling due to their widespread use in electric guitar and bass processing [4]. Traditional methods for VA rely on circuit-based approaches, such as Wave Digital Filters [5] and nodal analysis [6], which yield accurate results as they are provided with detailed knowledge of the internal circuitry. In recent years, data-driven black-box approaches leveraging deep learning have gained attention for their ability to emulate audio effects without explicit knowledge of circuit design. Neural networks trained on input-output audio have shown promising results in replicating complex effect behaviors [7, 8]. While high accuracy can be achieved, integrating control over effect parameters can be challenging. Conditioning methods have been proposed to allow parameter control in black-box neural models [9], requiring datasets with a large number of combinations of control parameters.

Beyond precise emulation of individual effects, only few recent studies have explored the interpolation capabilities of neural networks for blending multiple audio effects into new sonic textures or discovering new effects [10, 11, 12]. Moreover, limited interest has been devoted in the literature to the exploration of multiple versions of the same effect with different settings. In the context of audio effects, the sonic palette of analog overdrive pedals is rather limited, but different circuit design choices can lead to subtle differences in harmonic content and frequency response. Moreover, instead of conditioning a hybrid overdrive model with parameter and pedal information, we argue that a compact latent conditioning space can be used for seamless morphing among multiple effect parameters and circuits.

In this paper, we present an approach to the neural modeling of overdrive pedal effects conditioned with data from a cross-circuit and cross-setting latent space. The proposed approach models the behavior of multiple overdrive pedals at once, each with different settings. Cross-circuit/setting conditioning is achieved through a compact latent space obtained through a *latent-extractor* component, which consists of a Variational Autoencoder (VAE) with an adversarial component. The latent space is then used to condition a recurrent neural network that processes the input audio and generates the output signal. The processing network with conditioning was adapted from [9]. Our method is validated also thanks to the Parametric Overdrive pedal dataset (*pOD-set*), a newly collected open-source dataset, which spans across 27 analog overdrive pedals in multiple configurations, acquired using a custom-designed robot. Finally we propose a 2D user interface for controlling the modeling network in real-time.

The neural modeler is released as open-source (GPLv3 license) on GitHub, along with the code and hardware specifications for the recording robot [1]. The dataset is made available on Zenodo[2] under the Creative Commons Attribution-NonCommercial 4.0 License (CC BY-NC)[3]

The remainder of the paper is organized as follows. Section 2 provides an overview of works in neural effect modeling and representation learning. Section 3 describes the dataset and the recording robot. Section 4 presents the proposed method. Section 5 describes the experimental setup. In Section 6 we present and discuss the achieved results. Conclusions and final remarks are drawn in Section 7.

2. RELATED WORKS

2.1. Neural Effect Modeling

The emulation of audio effects through digital means is known as VA [1, 3]. Within VA, black-box approaches aim to match the out-

[1] https://github.com/domenicostefani/morphdrive
[2] https://doi.org/10.5281/zenodo.15389653
[3] https://creativecommons.org/licenses/by-nc/4.0/

put of a target device using functions or machine learning models without explicit knowledge of the internal circuitry [2, 13, 14, 15]. Black-box approaches based on neural networks have been gaining interest as of late, resulting in the development of various methods for neural modeling of effects [7, 9] and guitar amplifiers [4, 8, 16]. Useful resources on neural-network-based VA effect modeling are the review by Vanhatalo *et al.* [16] and the ongoing repository of audio-effects research by Comunità *et al.* [17].

The main deep learning architectures used in neural effect modeling are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and hybrid configurations [16]. Among the most prominent CNN-based approaches, Damskägg *et al.* [7] investigated real-time modeling of distortion pedals through a modified WaveNet architecture. A similar approach was used by Steinmetz *et al.* [18], who chose a modified WaveNet architecture (i.e., TCN) with large dilation factors and shallow configuration to model complex non-linear effects more efficiently. RNNs have been used in a number of VA approaches, including Long Short-Term Memory (LSTM) models [19] and Gated Recurrent Unit (GRU) models [4]. In [8], Wright *et al.* compared several recurrent architectures to WaveNet configuration, showing how LSTMs can provide lower processing speeds at the expense of different degrees of accuracy with highly non-linear effects. Instead, other hybrid approaches include the use of architectures such as convolutional autoencoders [20, 21].

Parameters and Conditioning: In neural effect modeling architectures, control parameters are integrated through model conditioning methods [22]. Recently, Yeh *et al.* [9] compared several conditioning methods for VA RNN architectures, evaluating their performance in modeling an overdrive pedal and an optical compressor. Conditioning methods require training with data recorded on a target device with many combinations of physical control parameters. To automate the recording of such data, Juvela *et al.* [23] proposed a data-collection pipeline for conditioned neural amplifier modeling, where relevant controls of a guitar amplifier were operated through electric motors.

Hybrid and New Neural Effects: Beyond attempts at precisely modeling existing audio effects, less interest has been devoted to exploring hybrid effects that can enable sounds in between existing effects, with a non-existent equivalent in the real world. Notably, Simionato *et al.* [10] proposed the use of neural networks for modeling multiple effects, resulting in a hybrid effect that can morph between audio effects continuously, exploiting the inherent interpolation capabilities of deep models. The authors trained an RNN with conditioning on a tube preamplifier, optical compressor, and tape recorder, providing qualitative consideration on the hybrid effect obtained. Steinmetz *et al.* [11] proposed a steering method based on training a model with fixed conditioning signals and analyzing the behavior with varying conditioning signals during inference. This work improved on the authors' previous study on overdrive networks with random weight initialization [24]. Naradowsky [12] proposed the use of VAEs for VA modeling of multiple guitar amplifiers. The author's preliminary findings hinted at the possibility of a single latent space being able to represent encodings of multiple timbre transformations. However, despite the strength of VAEs for interpolation between sounds, the author found the sound generation abilities of the VAE to be inferior to those of a WaveNet architecture. Conversely, the sole WaveNet did not perform well when trying to interpolate between existing effects. Nevertheless, while VA modeling of existing effects is a well-defined task with known quantitative evaluation procedures, discovering new effects or evaluating hybrid effects are open-ended tasks [10, 11].

2.2. Representation Learning

In modern deep learning, representation learning refers to the process of automatically retrieving meaningful features from raw data and encoding them in a compact space [25, 26]. By employing models to extract relevant information and reduce data dimensionality, such representations foster structured and interpretable feature embeddings, which can then be easily manipulated. In addition, models that learn non-discrete distributions enable coherent interpolation or random sampling in their latent spaces, generalizing over unseen variations in the data [27]. In this regard, VAEs proved particularly effective and have been widely exploited in neural synthesis, (i) facilitating direct resynthesis from sampled latents [28], (ii) promoting feature disentanglement across multiple descriptors in both supervised and unsupervised settings [29, 30], or (iii) enabling the concatenation of learned features with discrete labels [31]. With the growing popularity of deep generative models specialized in high-quality audio content, many studies incorporate latent representations as conditioning signals for larger models, leveraging their capabilities and complexity while granting flexible control over the generation process. For example, Liu and Jin [32] used an encoder to extract class-informed features for conditioning an RNN in an adversarial framework; similarly, Huang *et al.* [33] and Demerlé *et al.* [34] applied VAE latents within Latent Diffusion Models, while Rohnke *et al.* [35] leveraged them to condition a Parallel WaveNet. VAEs offer promising interpolation and clustering characteristics that have yet to be investigated for conditioning more capable VA modeling networks to enable continuous morphing among settings and effect types.

3. DATASET

We present *pOD-set*, a new dataset with 97 hours and 57 minutes of recordings covering 27 overdrive pedals, with 36 combinations of parameters for each. Out of the wide range of different overdrive pedals currently available on the market, we selected a subset of high-end, boutique pedals (see Tab. 1) - spanning from renowned classic circuits to novel, peculiar, and unconventional designs - which in our experience we believe may be representative of the broad spectrum of tonal nuances that characterizes modern guitar.

The different parameter settings represent combinations of the gain and tone knobs, typically found in most overdrive pedals. For each knob, we recorded six positions, covering their full range in evenly spaced increments (i.e., 0, 2, 4, 6, 8, and 10 on 10-mark scales). Additional controls, where present, were set to flat/neutral positions. Detailed settings are provided with the dataset.

The dataset was generated by processing a single input audio file through each parameter configuration of each analog pedal. The main input was a ~6-minute WAV file (48 kHz, 24-bit) from Yeh *et al.* [9], containing a diverse range of instrumental sounds. Moreover, we added three 4-second sinusoidal sweeps (15 Hz to 24 kHz) at different amplitudes (-6, -12, and -24 dBFS), separated by 400 ms of silence to capture the noise floor of each pedal. On each pedal, the volume knob was used to match the output level among different pedals, first manually adjusted to -3dBFS peak at the loudest setting. Since fine level matching is not feasible due to the circuit's nonlinear behavior across different parameters, we adopted per-class normalization of the recordings so that the

Table 1: *Analog overdrive pedals in* pOD-set.

	Label	Brand	Model
1	KOT	AnalogMan	King of Tone
2	HON	Bearfoot	Honey Bee
3	BEE	Beetronix	Overhive
4	WES	Bogner	Wessex MKII
5	BLU	Boss	BD-2-B50A
6	GLA	Cornerstone	Gladio
7	SS2	Cornish	SS2
8	ELE	Dr Scientist	The Elements
9	PLU	EarthQuaker Devices	Plumes
10	OCD	Fulltone	OCD v1.3
11	ZEN	Hermida Audio	Zendrive
12	PRE	Horizon Devices	Precision Drive
13	TUB	Jam Pedals	Tube Dreamer
14	KTR	Klon	KTR
15	ACQ	Lichtlaerm Audio	Acquaria
16	FEL	Lunastone	Big Fella
17	CHI	Pettyjohn	Chime
18	LIG	Rawkworks	Light OD
19	ZOI	SviSound	Overzoid
20	DUM	Tanabe	Dumkudo
21	RUM	Tone City	Big Rumble
22	JAN	Vemuram	Jan Ray
23	SIL	Vox	Silk Drive
24	RED	Way Huge	Red Llama[4]
25	MOF	Wampler	Mofetta
26	385	Walrus Audio	385 MKII
27	SOU	Xotic	Soul Driven

loudest peak for each pedal was precisely -3 dBFS. Normalization was minimal, with an average amplification of 0.33 $\pm$ 0.87 dB. Complete normalization details for each file are provided with the dataset.

All dataset samples were recorded using a MOTU M4 audio interface. The input signal was routed through a Radial ProRMP passive re-amp box to match the pedals' input impedance. We measured and removed the total signal chain latency of 1268 samples from each file for accurate alignment with the input file. Finally, since signal phase is *per se* tonally irrelevant, we inverted the polarity for phase-inverting circuits to maintain consistency across the dataset.

3.1. Robotic Parameter Controller

To ensure precise and consistent parameter settings across pedals, we built a parameter-controller robot inspired by [23]. The robot is composed of off-the-shelf, inexpensive components, including two stepper motors (28BYJ-48), an Arduino "Uno" microcontroller, and a velcro platform for securing pedals. Each motor controls the shaft of an individual potentiometer via a toothed belt and two identical pulleys. To ensure proper tensioning and avoid obstructions (e.g., additional knobs on the pedal), the motors are mounted on raised, laser-cut acrylic rails (see Fig. 1).

The 28BYJ-48 stepper motors have a step of 5.625 degrees and an internal gear reduction with a 1/64 ratio, resulting in a total of 4096 steps per revolution and a step of 0.088º. The Arduino board handles USB serial commands from a computer, instructing the

[4]Modified with additional tone control.

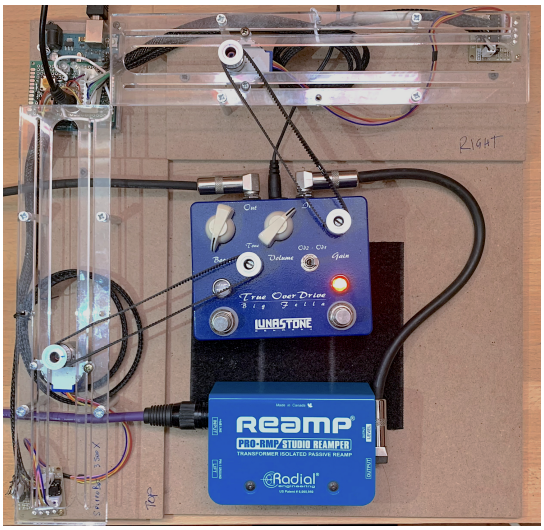

Figure 1: *Recording robot with one of the pedals connected.*

robot to move one or both potentiometers to a specific position. A PureData patch automates the entire recording process, including playing the input audio file, recording the output, renaming files based on the current pedal and setting, and sending commands to the Arduino to adjust the motors for the next position. For each pedal, the setup took approximately one minute, followed by 3 hours and 40 minutes of data recording. The Arduino code, PureData recorder patch, and hardware designs are available on the project's repository[1].

4. METHOD

The proposed pipeline is divided into two functional blocks: (i) a *latent extractor* and (ii) a *processing network* (see Fig. 2). The latent extractor encodes the behavior of each circuit and parameter configuration into a highly-compressed representation, i.e., a conditioning vector. In turn, the processing network is responsible for generating the output audio signal, given the input audio signal and the conditioning vector.

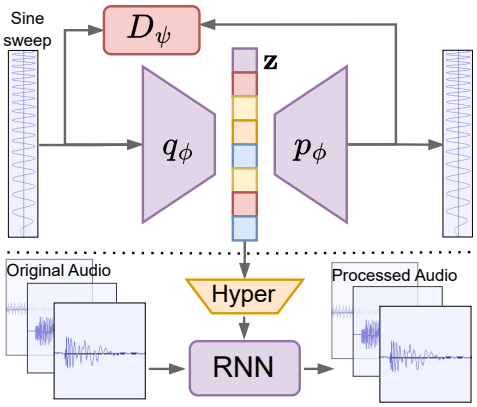

Figure 2: *System architecture, comprising the latent extractor VAE and the processing network with conditioning.*

4.1. Latent Extractor - VAE

The latent space extractor is a VAE enforced with an adversarial component. The VAE is trained on sinusoidal frequency sweeps from multiple pedals and parameter configurations. The model consists of a 1D convolutional encoder q_ϕ and of a quasi-symmetrical decoder p_θ. Both are 10 layers deep, with p_θ incorporating a final denoising layer with sinusoidal activation. q_ϕ extracts an 8-dimensional unimodal latent representation $\mathbf{z}$ from an input signal $\mathbf{X}$, such that $\mathbf{z} = q_\phi(\mathbf{X}) \sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$; p_θ then reconstruct the input $\hat{\mathbf{X}} = p_\theta(\mathbf{X}|\mathbf{z})$. The training objective is to maximize the Evidence Lower Bound (ELBO), balancing reconstruction accuracy and latent space regularization:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{X})} \left[\log p_\theta(\mathbf{X}|\mathbf{z})\right] - D_{\text{KL}} \left(q_\phi(\mathbf{z}|\mathbf{X}) \| p(\mathbf{z})\right) \quad (1)$$

where the first term is the reconstruction loss, and the second one corresponds to the Kullback-Leibler (KL) divergence, which regularizes the latent space. As for the reconstruction loss, we adopted a weighted sum of three terms: Mean Squared Error (MSE), Huber Loss, and Multi-Resolution Short-Time Fourier Transform (MR-STFT) Loss [36], capturing both time-domain and frequency-domain fidelity. The latter is defined as:

$$\mathcal{L}_{\text{STFT}} = \sum_{w \in W} \left(\|STFT_w(x) - STFT_w(\hat{x})\|_1\right) \quad (2)$$

where $STFT_w(x)$ denotes the Short-Time Fourier Transform computed at different window sizes $w \in W$. To further improve reconstruction results, we exploited adversarial training, introducing a convolutional discriminator D_ψ. During training, we alternate between optimizing D_ψ and the VAE in a minimax game. In the first stage, D_ψ is trained to identify real samples $p_{\mathbf{X}}$ and reconstructed ones $p_{\hat{\mathbf{X}}}$ using a BCE loss:

$$\mathcal{L}_{D_\psi} = -\mathbb{E}_{\mathbf{X} \sim p_{\mathbf{X}}} \left[\log D_\psi(\mathbf{X})\right] - \mathbb{E}_{\hat{\mathbf{X}} \sim p_{\hat{\mathbf{X}}}} \left[\log(1 - D_\psi(\hat{\mathbf{X}}))\right] \quad (3)$$

In a second training stage, we update the VAE adding to the term in Eq. (1) an adversarial loss weighted by a fixed γ:

$$\mathcal{L}\text{adv} = -\gamma \mathbb{E}\hat{\mathbf{X}} \sim p_{\hat{\mathbf{X}}} \left[\log D_\psi(\hat{\mathbf{X}})\right] \quad (4)$$

4.2. Processing Network - RNN

The processing network is trained on input and output audio for multiple pedals and configurations, and the relative conditioning vectors from the latent extractor. We employed three processing networks relying on the implementation provided by Yeh *et al.* [9]. The authors proposed several networks and experimented with different conditioning mechanisms; out of these, we choose to use GRU cells combined with two hypernetworks, namely *StaticHyper* and *DynamicHyper* as they represent a good compromise between efficiency and signal quality. The purpose of such hypernetwork is to generate weight matrices for the main RNN. For conciseness, here we briefly summarize the key concepts behind such hypernetworks while referring the reader to the original paper for detailed formulations.

The StaticHyper mechanism is the most computationally efficient, as it generates fixed weight matrices $\mathbf{W}_x$, $\mathbf{W}_h$, and the bias vector $\mathbf{b}$ by passing a conditioning vector $\mathbf{c}$ into a Multi-Layer Perceptron (MLP) φ, which outputs the corresponding projections:

$$\mathbf{W}_x = \varphi_x(\mathbf{c}), \quad \mathbf{W}_h = \varphi_h(\mathbf{c}), \quad \mathbf{b} = \varphi_b(\mathbf{c}) \quad (5)$$

On the other hand, the DynamicHyper mechanism computes weight matrices at each timestep using a second recurrent network, which takes as input the concatenation of both $\mathbf{c}$ and the previous hidden state h^γ_{t-1} from the GRU (denoted as γ). The produced transformations d_h, d_z and the features $\mathbf{z}_h$, $\mathbf{z}_x$ are then integrated in the processing GRU itself via element-wise multiplication:

$$h^\gamma_t = \tanh(d_h(\mathbf{z}_h) \odot \mathbf{W}^\gamma_h h^\gamma_{t-1} + d_z(\mathbf{z}_x) \odot \mathbf{W}^\gamma_x x^\gamma_t) \quad (6)$$

Following the original implementation, the training objective combines a weighted MR-STFT - Eq. (2) - and an MSE loss:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{STFT}} + \mathcal{L}_{\text{MSE}} \quad (7)$$

5. EVALUATION

In this section, we describe the experimental setup and present a simplified user interface for exploring the network's latent space.

5.1. Experimental Setup

We conduct a series of experiments with the proposed architecture and our *pOD-set*.

Setup 1: we train the latent extractor (VAE) on the full dataset in an unsupervised manner. The model learns to reconstruct a chunk of the sine sweeps in the dataset (1200 Hz - 6600 Hz), resampled at 32 kHz to balance efficiency and representation quality. Each setting is encoded into an 8-dimensional latent vector; preliminary experiments with lower-dimensional spaces resulted in poor reconstruction performance. We use stratified K-fold cross-validation (K=5), running for 5000 epochs, with a batch size of 32 and Adam optimizer (initial learning rate 10^{-3}, halved every 500 epochs). After training, we extract the latent vectors for all samples and store them for visualization.

Setup 2: we select three progressively larger subsets of pedals prioritizing diversity in timbral characteristics:

- **2 pedals**: {`HON`, `ZEN`}
- **4 pedals**: {`HON`, `ZEN`, `FEL`, `SOU`}
- **8 pedals**: {`HON`, `ZEN`, `FEL`, `SOU`, `KOT`, `RED`, `SS2`, `CHI`}

For each subset, we train the latent extractor as in the previous setup and use the extracted latent as conditioning signals for two processing networks, namely DynamicHyper-RNN (D-GRU) and StaticHyper-RNN (S-GRU), comparable in size to the models in [9]. Finally, we evaluate the system capability in a real-time setting with a smaller variant of the S-GRU, denoted as Real-time StaticHyper-RNN (RT-S-GRU). For all RNNs, both the input and processed audio files are segmented into 4096-sample chunks. Preliminary experiments showed that the network converges quite quickly without requiring a large amount of data. To optimize training efficiency, we therefore retain only the first 40% of each sample in the dataset and train the model for 3 epochs, with a batch size of 128 and Adam optimizer with an initial learning rate of 10^{-3} halved every 500 steps. Results from these experiments are discussed in Section 6.

5.2. 2D Control GUI

Despite the 8-dimensional latent space enabling smooth morphing across different control configurations and overdrive effects, such a high-dimensional control space may be arduous to navigate for users. We propose a 2D graphical user interface (GUI) (see Fig. 3)

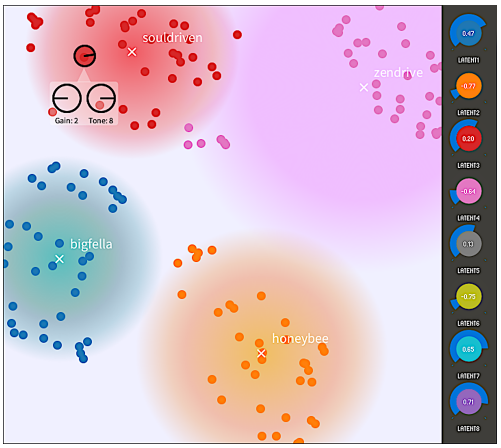

Figure 3: *2D latent space navigation GUI. The red cursor moves with the user's mouse, allowing them to move across the 2D reduction of the 8D conditioning space. Different settings for each pedal are represented by colored dots. The 8 knobs (right) represent single dimensions of the conditioning vector.*

that represents the 8-dimensional conditioning space through dimensionality reduction with t-distributed stochastic neighbor embedding (t-SNE). Different color points represent training configurations, and users can navigate the space with the cursor. A 4-layer MLP is trained to reverse the dimensionality reduction, predicting an 8-value vector for each point on the 2D plane. We train the model for 1500 epochs, with a batch size of 32 and Adam optimizer and a fixed learning rate of 10^{-3}, achieving validation MSE $< 6 \times 10^{-2}$. Additionally, the 8 conditioning values are presented on the side for both visualization and control, compensating for the dimensionality lost in the reduction.

6. RESULTS AND DISCUSSIONS

6.1. Latent Space Extraction

The reconstruction performance of the latent extractor (i.e., VAE) was measured in terms of MSE and MR-STFT, as mentioned in Section 5.1. The results are shown in the upper part of Tab. 2. Due to the varying characteristics of the circuits, we occasionally observed slight fluctuations in the reconstruction metrics and the structure of the latent spaces depending on the subset considered. While an in-depth analysis of all possible combinations within our dataset is beyond the scope of this work, the reported metrics still offer a meaningful assessment of the latent extractor.

Since the VAE is trained in an unsupervised manner and is fed data with consistent pitch (i.e., processed sine sweeps), the extracted latent spaces are arranged according to the sole amplitude and timbral characteristics of each pedal and parameter configuration. The 2D projection of the latent space extracted from the whole dataset (see Fig. 4) reveals distinct clusters for pedals with peculiar sonic traits (e.g., `ZOI`, `BEE`, or `385`), while pedals with similar circuit designs (e.g., `LIG` and `KTR`, or `JAN` and `GLA`) tend to lie closer or overlap. We consider this as a positive asset of our approach, as it allows the model to autonomously correlate similarities, facilitating the learning process for the processing networks. Previous attempts at using supervised learning resulted in satisfactory reconstruction metrics and produced well-separated clusters of individual pedals, but the too-condensed clusters turned less informative for the RNNs.

This behavior is evident in the 2D reductions of the latent spaces extracted for the 2, 4, and 8-pedal configurations (see Fig. 5). All plots show not only a good separation of clusters relative to different pedals but also a distribution spread across the whole space. Moreover, further analysis revealed that configurations with similar gain and tone settings were positioned closer to each other in the latent space.

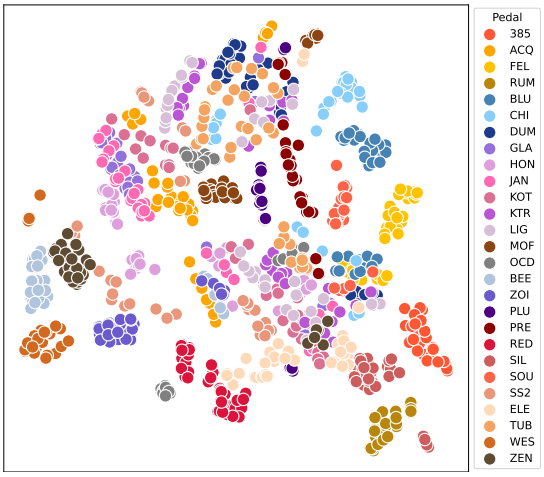

Figure 4: *2D t-SNE of the 8-dimensional latent space extracted with the VAE from the entire dataset.*

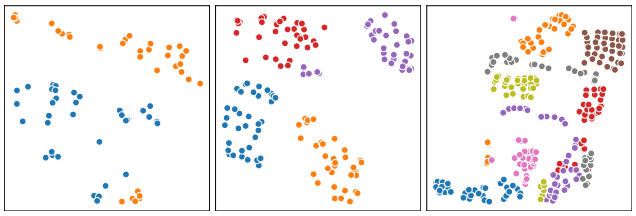

Figure 5: *2D t-SNE reductions of the latent space for the 2, 4, and 8-pedal configurations.*

6.2. Audio Processing

Results of the three processing networks (i.e., D-GRU, S-GRU, and RT-S-GRU) across the two metrics considered are provided in Tab. 2. Overall, the RNNs proved robust in correctly reconstructing the processed waveforms and achieved state-of-the-art results. Despite the increased complexity, the 8-dimensional conditioning signal proved sufficiently informative to allow the network to effectively process the input signal while maintaining perceptual coherence. In all cases, the 4-pedal configuration yielded the best results, highlighting this setup as the best compromise between the meaningfulness of the latent space, complexity, and amount of data for training. The 2- and 8-pedal configurations exhibit slightly worse results, with a large spike in the 8-pedal RT-S-GRU, likely due to its small number of parameters.

To provide a comparison with the original method [9] we based ours on, we also train the whole pipeline on a single pedal - `HON`, for consistency with the other subsets. As presented in Tab. 3, our latent-conditioned D-GRU and S-GRU outperforms

Table 2: *Metrics for the latent extractor (VAE) and all the processing networks (D-GRU, S-GRU, RT-S-GRU). The lowest error for each network is in bold.*

Model	Params	Pedals	↓ MSE $*10^{-3}$	↓ MR-STFT
VAE	5,292,396	2	0.548	0.238
		4	0.346	0.168
		8	0.531	0.193
		27	**0.312**	**0.154**
D-GRU	20,769	2	0.409	0.429
		4	**0.349**	**0.411**
		8	0.533	0.642
S-GRU	57,569	2	0.61	0.495
		4	**0.408**	**0.439**
		8	0.569	0.599
RT-S-GRU	1,849	2	1.384	1.113
		4	**0.836**	**0.814**
		8	2.358	1.469

Table 3: *Comparison of processing networks trained on a single pedal (HON) with existing literature [9].*

Model	↓MAE (L1) $*10^{-2}$		↓ MR-STFT	
	Ours	Yeh *et al.*	**Ours**	Yeh *et al.*
D-GRU	**0.763**	15.0	**0.251**	0.428
S-GRU	**0.838**	1.7	**0.329**	0.698

the original models conditioned on knob configuration. While the architecture of the S-GRU implies an increase of the number of parameters (ours 57,569 against the original 30,369), the size of the D-GRUs are instead comparable (20,769 against 20,289). Despite the inclusion of all knob combinations (36 against 25) potentially contribute to the improved performance, the generalization capabilities of the RNNs still suggests that latent conditioning provides a more informative representation than a discrete, knob-based one. Further experiments on the architecture, possibly including the same pedal modeled by the authors (Boss OD-3), may better support this claim.

6.3. Latent Space Exploration and Hybrid Effects

Beyond the evaluation of individual components of the architecture based on the real effect configurations measures, we are interested in assessing the entire modeling network as a new hybrid effect to understand what happens in-between "real" settings. For these experiments, we employ the 4-pedal configuration, as it yielded the best results.

First, we sample an 8-dimensional conditioning space and perform value sweeps on each individual dimension. Fig. 6 shows the effect on an input sinewave cycle of such sweeps along the first four dimensions of the conditioning vector. Some dimensions appear to be mostly related to the amount of amplification and compression (e.g., dimension 2), while others affect various distortion and filtering nuances. However, assigning a clear perceptual/parametric meaning to each individual dimension remains challenging: this is a problem common to many works on the discovery of new neural effects [24, 11] or hybrid effects [10].

The complexity of such evaluation motivates our 2D interface as the lower dimensionality of control can enable visual understanding of the topography of the space. To do so, we retrieved conditioning vectors for every 2D point in a 100 × 100 grid, used them for the S-GRU processing a 440Hz sinewave, and finally performed an analysis of the distortion, compression, and filtering

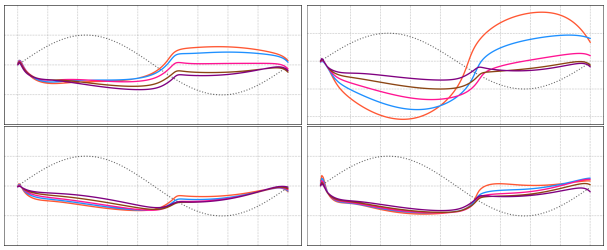

Figure 6: *Effect on a sinewave cycle (gray) of independent value sweeps along the first four dimensions of the conditioning space (Values: {-1, -0.5, 0, 0.5, 1}). Phase was inverted for visualization.*

characteristics of each output. Distortion is measured as the Total Harmonic Distortion (THD), compression as the Crest factor (Eq. (8) [37]), and filtering is assessed through the spectral centroid [38] of the output signal.

$$CrestFactor(dBFS) = Peak(dBFS) - RMS(dBFS) \quad (8)$$

The combination of the 2D interface and the aforementioned metrics allows us to get a general understanding of the arrangement of tonal characteristics over the latent space.

The THD map (Fig. 7, left) shows "islands" of high distortion that exist across multiple groups of pedals. Moreover, all the pedal clusters contain points spanning from higher to lower distortion areas. Interestingly, the islands with the highest THD do not contain any real sample, indicating that the combination of the processing network and 2D-to-8D mapping network may develop their own extreme behavior that steers from known data. Further visual analysis of outputs from these areas showed peculiar characteristics, e.g., second harmonics being more pronounced than the fundamental frequency.

The crest factor (Fig. 7, center) shows quite different areas for each pedal. For the `ZEN` pedal, the crest factor shows the highest compression, which is consistent with the specific pedal's known behavior. Other pedals, e.g., `HON`, show more varied crest values for each setting. The two lowest compression areas (light blue) correspond to the highest THD areas. While the leftmost low compression area seems to originate from a large number of settings from the `FEL` pedal, the rightmost area is far from all real settings. This last area supports the hypothesis of emergent independent behaviors of the entire conditioning and processing pipeline, which develops an overall novel sound effect.

Finally, the spectral centroid map (Fig. 7, right) shows each pedal cluster having an overall homogeneous centroid value, which differs among clusters. This could reflect the different setting-invariant tonal nature of the various circuits, which can originate from fixed parts of the respective circuits, such as the input filtering stage. Areas between real configurations are harder to interpret, but the area with the highest centroid roughly coincides with that having the highest THD. The magnitude of the centroid in this area supports the aforementioned emerging behavior.

This analysis on the 2D interface represents a first step towards a quantitative analysis of new and hybrid effects found in between real points of the conditioning space. Furthermore, any of these maps could be provided underneath the 2D interface to brief users on the characteristics of the space, bringing back some of the confidence that can be lost with the removal of the direct gain and tone controls.

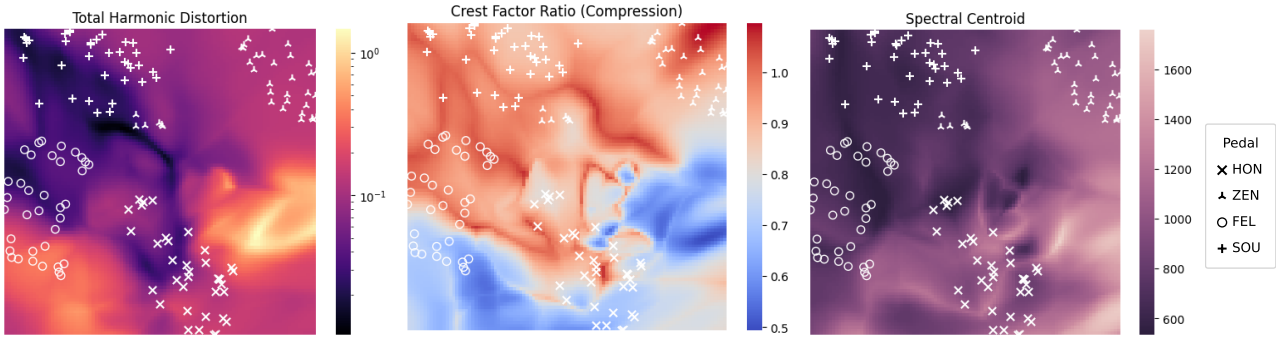

Figure 7: *Analysis of a) total harmonic distortion; b) crest factor; and c) spectral centroid, across the 2D latent reduction with the 4-pedal configuration (see Fig. 3).*

7. CONCLUSIONS

In this paper, we presented a method for neural effect modeling based on compact latent space conditioning, which enables continuous morphing between effect circuits and settings. Moreover, we introduced *pOD-set*, a new dataset of analog overdrive pedal recordings with 36 combinations of gain and tone parameters. The proposed modeling network exploited a VAE to construct an 8D latent space encoding timbral characteristics of multiple circuits and settings. The latent space was then used as conditioning signal for several hyper-recurrent models, which process audio accordingly. The 8D latent conditioning outperformed the 2D discrete approach in single-pedal configurations, while remaining robust across the 2, 4, and 8-pedal settings. We further proposed a 2D interface for navigating the latent space, leveraging dimensionality reduction and a MLP to reverse it. The interface proved useful in exploring and evaluating the networks beyond the accuracy in modeling real settings, as it allowed us to assess distortion, compression, and filtering characteristics in areas between existing ones. As a result, we found behaviors that confirmed characteristics of existing pedals, as well as emerging behaviors to be attributed to the entire conditioning and processing pipeline.

Handling multiple pedals simultaneously may lead to a slight reduction in the processing accuracy; however, such trade-off allows for increased flexibility. Indeed, we argue that our primary goal is not to perfectly emulate specific configurations, but to balance realistic behavior with the creation of novel, unconventional effects. Still, future work will look at optimizing the latent space to balance complexity and separation.

We strongly encourage creative uses of the dataset in the hope that it will be useful for the community to explore the tonal characteristics of analog overdrive pedals.

8. ETHICAL STATEMENT

All product names and trademarks mentioned in this paper are the property of their respective owners. The inclusion of any pedal in *pOD-set* does not imply any endorsement, sponsorship, or licensing agreement with the manufacturer. The dataset was created independently using standard recording techniques, and no reverse engineering of circuits or proprietary algorithms was performed. The pedals included in the dataset were purchased through regular retail channels, and the recordings document their performance under controlled conditions. The dataset is distributed to be used for research purposes.

9. ACKNOWLEDGMENTS

The authors would like to thank Gregorio A. Giudici for the help on the real-time scripts and Elisa Pisetta for granting us access to part of her private collection.

10. REFERENCES

[1] J. O. Smith, *Physical Audio Signal Processing for Virtual Musical Instruments and Digital Audio Effects*, W3K Publishing, 2010.

[2] M. J. Kemp, "Analysis and simulation of non-linear audio processes using finite impulse responses derived at multiple impulse amplitudes," in *The 106th AES Convention*, 1999.

[3] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," in *DAFX: Digital Audio Effects*, U. Zölzer, Ed., pp. 473–522. Wiley Online Library, 2011.

[4] A. Wright, E.-P. Damskägg, and V. Välimäki, "Real-time black-box modelling with recurrent neural networks," in *Int. Conf. on Digital Audio Effects (DAFx19)*. University of Birmingham, 2019.

[5] A. Fettweis, "Wave digital filters: Theory and practice," *Proc. of the IEEE*, vol. 74, no. 2, pp. 270–327, 1986.

[6] D. Yeh, *Digital Implementation of Musical Distortion Circuits by Analysis and Simulation*, Ph.d. thesis, Stanford University, June 2009.

[7] E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time modeling of audio distortion circuits with deep learning," in *Proc. of the SMC Conf.* SMC, 2019, pp. 332–339.

[8] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Applied Sciences*, vol. 10, no. 3, 2020.

[9] Y.-T. Yeh, W.-Y. Hsiao, and Y.-H. Yang, "Hyper recurrent neural network: Condition mechanisms for black-box audio effect modeling," in *Proc. 27th Int. Conf. on Digital Audio Effects (DAFx24)*, Sept. 2024, pp. 97–104.

[10] R. Simionato and S. Fasciani, "Hybrid neural audio effects," in *Proc. of the SMC Conf.* SMC, 2024.

[11] C. J. Steinmetz and J. D. Reiss, "Steerable discovery of neural audio effects," in *5th Workshop on Machine Learning for Creativity and Design at NeurIPS*, 2021.

[12] J. Naradowsky, "Amp-space: A large-scale dataset for fine-grained timbre transformation," in *Proc. 24th Int. Conf. on Digital Audio Effects (DAFx21)*, 2021, pp. 57–64.

[13] D. J. Gillespie and D. P. W. Ellis, "Modeling nonlinear circuits with linearized dynamical models via kernel regression," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[14] A. Novak, L. Simon, P. Lotton, and J. Gilbert, "Chebyshev model and synchronized swept sine method in nonlinear audio effect modeling," in *Proc. Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 2010, pp. 423–426.

[15] S. Orcioni, A. Terenzi, S. Cecchi, F. Piazza, and A. Carini, "Identification of volterra models of tube audio devices using multiple-variance method," *J. Audio Eng. Soc.*, vol. 66, no. 10, pp. 823–838, Oct. 2018.

[16] T. Vanhatalo, P. Legrand, M. Desainte-Catherine, P. Hanna, A. Brusco, G. Pille, and Y. Bayle, "A review of neural network-based emulation of guitar amplifiers," *Applied Sciences*, vol. 12, no. 12, 2022.

[17] M. Comunita and J. Reiss, "Afx-research: an extensive and flexible repository of research about audio effects," Available at https://doi.org/10.5281/zenodo.13380393, 2024.

[18] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time modeling of analog dynamic range compression," in *152nd AES Convention*, 2022, https://arxiv.org/abs/2102.06200.

[19] Z. Zhang, E. Olbrych, J. Bruchalski, T. J. McCormick, and D. L. Livingston, "A vacuum-tube guitar amplifier model using long/short-term memory networks," in *SoutheastCon 2018*, 2018, pp. 1–5.

[20] M. A. Martínez Ramírez, E. Benetos, and J. D. Reiss, "Deep learning for black-box modeling of audio effects," *Applied Sciences*, vol. 10, no. 2, 2020.

[21] M. A. Martínez Ramírez and J. D. Reiss, "Modeling nonlinear audio effects with end-to-end deep neural networks," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 171–175.

[22] R. Simionato and S. Fasciani, "Conditioning methods for neural audio effects," in *Proc. of the SMC Conf.* SMC, 2024.

[23] L. Juvela, E.-P. Damskägg, A. Peussa, J. Mäkinen, T. Sherson, S. I. Mimilakis, K. Rauhanen, and A. Gotsopoulos, "End-to-end amp modeling: from data to controllable guitar amplifier models," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[24] C. J. Steinmetz and J. D. Reiss, "Randomized overdrive neural networks," in *4th Workshop on Machine Learning for Creativity and Design at NeurIPS*, 2020.

[25] I. C. Kaadoud, L. Fahed, and P. Lenca, "Explainable ai: a narrative review at the crossroad of knowledge discovery, knowledge representation and representation learning," in *MRC 2021: 12th Int. Workshop Modelling and Reasoning in Context*, 2021, vol. 2995, pp. 28–40.

[26] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, "Self-supervised representation learning: Introduction, advances, and challenges," *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 42–62, 2022.

[27] N. Bryan-Kinns, B. Zhang, S. Zhao, and B. Banar, "Exploring variational auto-encoder architectures, configurations, and datasets for generative music explainable ai," *Machine Intelligence Research*, vol. 21, no. 1, pp. 29–45, 2024.

[28] K. Tatar, K. Cotton, and D. Bisig, "Sound design strategies for latent audio space explorations using deep learning architectures," in *Proc. of the SMC Conf.*, Stockholm, Sweden, 2023, pp. 239–246.

[29] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using Gaussian mixture variational autoencoders," in *Proc. of the 20th Int. Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 746–753.

[30] K. Tanaka, Y. Bando, K. Yoshii, and S. Morishima, "Unsupervised disentanglement of timbral, pitch, and variation features from musical instrument sounds with random perturbation," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2022, pp. 709–716.

[31] A. V. Puche and S. Lee, "Caesynth: Real-time timbre interpolation and pitch control with conditional autoencoders," in *31st Int. Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2021, pp. 1–6.

[32] Y. Liu and C. Jin, "ICGAN: An implicit conditioning method for interpretable feature control of neural audio synthesis," in *Proc. 27th Int. Conf. on Digital Audio Effects (DAFx24)*, Guildford, United Kingdom, September 2024, pp. 73–80.

[33] H. Huang, J. Man, L. Li, and R. Zeng, "Musical timbre style transfer with diffusion model," *PeerJ Computer Science*, vol. 10, pp. e2194, 2024.

[34] N. Demerlé, P. Esling, G. Doras, and D. Genova, "Combining audio control and style transfer using latent diffusion," in *Proc. of the 25th Int. Society for Music Information Retrieval Conference (ISMIR)*, 2024, pp. 721–728.

[35] J. Rohnke, T. Merritt, J. Lorenzo-Trueba, A. Gabrys, V. Aggarwal, A. Moinet, and R. Barra-Chicote, "Parallel wavenet conditioned on vae latent vectors," *arXiv preprint arXiv:2012.09703*, 2020.

[36] J. O. Smith, III, *Spectral Audio Signal Processing*, W3K Publishing, 2011, Section: Multiresolution STFT.

[37] M. Wendl and H. Lee, "The effect of dynamic range compression on loudness and quality perception in relation to crest factor," in *Audio Engineering Society Convention 136*. Audio Engineering Society, 2014.

[38] E. Schubert and J. Wolfe, "Does timbral brightness scale with frequency and spectral centroid?," *Acta acustica united with acustica*, vol. 92, no. 5, pp. 820–825, 2006.