PREPRINT, Published DOI: https://doi.org/10.17743/jaes.2026.0278

# *Prism*: Neural Modeling of Multiband Mixtures of Nonlinear Analog Effects

**FRANCESCO ARDAN DAL RÍ, DOMENICO STEFANI,**
(francesco.dalri-2@unitn.it) (domenico.stefani@unitn.it)
**LUCA TURCHET,** *AES Member*, **and NICOLA CONCI**
(luca.turchet@unitn.it) (nicola.conci@unitn.it)

*DISI - Department of Information Engineering and Computer Science, University of Trento, Italy*

This paper introduces *Prism*, a neural audio modeling framework for real-time emulation of multiband mixtures of nonlinear analog effects, including overdrive, distortion, and fuzz pedals. While conventional approaches involve models specifically trained to emulate the response of single devices on the whole frequency spectrum, *Prism* exploits band-aware conditioning, allowing independent end-to-end processing of multiple frequency bands directly on the input waveform. The model leverages latent representations to modulate a processing network, supporting simultaneous real-time emulation of multiple effect types and complex inter-band interactions. Prism is evaluated through benchmarks on audio quality and computational efficiency, yielding performance comparable with state-of-the-art virtual analog models despite the broader scope of the task. *Prism* allows extended control on hybrid and creative audio manipulation beyond conventional virtual analog modeling.

## 0 INTRODUCTION

Virtual Analog (VA) modeling refers to replicating the sound and behavior of analog audio equipment using digital signal processing [1]. Audio effect units, in both hardware and software form, transform audio signals, introducing effects such as reverberation, distortion, and modulation. Among these, distortion, overdrive, and compression are known for their nonlinear input-output behavior.

Modeling of analog audio effect units is a popular research area. Conventionally, VA has relied on white-box approaches, i.e., using detailed circuit analysis and component-level simulations [2, 3, 4]; recent advances in machine learning have made it possible to have real-time gray- and black-box modeling approaches that require limited or no prior knowledge of the underlying analog circuit and inter-component relationships [5, 6, 7, 8].

Besides the most common objective of precisely modeling individual units, some recent efforts have focused on the use of black-box approaches for creating new and hybrid audio effects [9, 10, 11]. These works explored the use of conditioning signals for deep learning systems to enable blending of different effects, relying on the interpolation capabilities of neural networks to enable new sound processing capabilities.

A promising direction for future research is granting audio professionals control over semantically meaningful parameters of these hybrid effects, bridging the gap between the sonic possibilities of current experimental models and the practical need for intuitive controls. Among these possibilities, multiband processing is a promising approach. Typically used in mastering and mixing applications [12], multiband processing involves splitting the signal into different frequency bands for separate processing, allowing for precise control over the tonal balance and dynamics of the sound [13].

This work introduces *Prism*, a framework for neural modeling of multiband mixtures of nonlinear analog effects - including overdrive, fuzz, and distortion - by directly learning the full-band transfer function. *Prism* therefore enables hybrid effect behavior where frequency bands may be processed as either of the modeled effects with different parameters. Band-specific latent representations of multiple settings are used as conditioning signals, enabling independent control of nonlinear behavior across frequency regions. However, audio is processed in a single forward pass, increasing efficiency and real-time performance. *Prism*'s design supports both the simultaneous emulation of different effects and creative exploration of hybrid and cross-band interactions, not achievable with conventional single-band, single-device systems.

The remainder of the paper presents architectures, pipelines, and evaluations across multiple configurations. Code and audio samples are made available online[1].

[1] https://github.com/return-nihil/Prism

PREPRINT, Published DOI: https://doi.org/10.17743/jaes.2026.0278

# 1 BACKGROUND

## 1.1 Nonlinear Neural Audio Modeling

In the context of VA, black-box approaches aim at emulating the response of nonlinear artifacts by relying solely on input-output measurements [5, 6], without explicit knowledge of the specific effect internal circuitry [7, 8]. Neural networks currently constitute the dominant approach for black-box VA [5, 14, 15], and several methods have been proposed to digitally reproduce various nonlinear devices such as distortions [16, 17], amplifiers [18, 19], and compressors [20, 21].

Convolutional neural networks and recurrent neural networks lie among the most widespread architectures [18]. For instance, Damskägg *et al.* [22] demonstrated that a modified WaveNet architecture can successfully model guitar distortion pedals in real-time; Steinmetz and Reiss [23] achieved similar results using a Temporal Convolution Network (TCN) with large dilation factors; Wright *et al.* [6] modelled a digital amplifier and a distortion pedal using both Long-Short Time Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks. Hybrid strategies for improving and optimizing models' capabilities have also been developed. For example, Ramirez *et al.* [14] deployed convolutional AutoEncoders (AEs) for feature extraction and latent representations, capturing complex nonlinearities while reducing model size; Huhtala *et al.* [24] proposed the use of a Koopman-Linearised Audio Neural Network to lift audio signals into high-dimensional spaces for more efficient linear prediction.

With these models showing almost-indistinguishable perceptual quality in the emulation of such pieces of hardware, recent work began to explore their potential beyond conventional analog paradigms. Relevant examples include Steinmetz and Reiss' Randomized overdrive Neural Network [25], exploiting randomly-initialized TCNs to create unpredictable distortion effects; the same authors [11] subsequently applied a similar architecture in a steerable configuration to interpolate between different types of effects, e.g., compression and reverb. Similarly, Dal Rí *et al.* [9] exploited latent spaces from a Variational AutoEncoder (VAE) trained on multiple overdrives as a conditioning signal for a GRU processing network, allowing for novel, in-between settings.

## 1.2 Multiband Processing

A fundamental technique in audio engineering, Multiband Processing involves the split of a signal into multiple frequency bands, each treated independently before final mixdown. This method enables fine-grain control over specific spectral content, and is widely used in audio production [12].

Neural networks are not new to multiband environments: several works exploit band-wise representations for audio tasks such as classification [26], source separation [27], or synthesis [28]. These approaches operate on spectral inputs or intermediate band features instead of learning full-band nonlinear transformations.

Existing multiband neural solutions in the realm of VA mostly follow a parametric strategy, which is training deep models to generate control parameters for external digital processors, rather than act as processors themself. For instance, Mimilakis *et al.* [29] and Ioniță *et al.* [30] explored multiband compression, respectively predicting gain/exponent factors and ratio, and threshold parameters for the processor; Sarkar and Lindborg [31] presented a model to estimate per-band control parameters for a parametric equalizer.

# 2 METHODOLOGY

The architecture of *Prism* consists of a VAE model, extracting latent representations as conditioning signals, and of a processing network to transform input audio accordingly. VAE conditioning for hybrid effects is inspired by [9]. The processing network builds upon both causal TCN [32] and WaveNet [33] architectures, and integrates band-aware conditioning and Feature-wise Linear Modulation (FiLM) [34]. An overview of the proposed architecture is summarized in Fig. 1, and presented in detail in the following sections.

## 2.1 Architectures

### *2.1.1 Latent Extractor*

The latent extractor is a 1D convolutional variational encoder-decoder architecture. The encoder $q_\phi$ produces a latent representation $\mathbf{z} \in \mathbb{R}^{d_c}$ with $d_c = 8$ from the input sine sweep $\mathbf{sw} \in \mathbb{R}^{T\times 1}$, where $T$ denotes the number of samples, such that $\mathbf{z} = q_\phi(\mathbf{sw}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$, while a decoder $p_\theta$ reconstructs the input $\hat{\mathbf{sw}} = p_\theta(\mathbf{sw}|\mathbf{z})$. The two modules are symmetrical, each composed of convolutional blocks, with LeakyReLU activations in $q_\phi$ and Sigmoid Linear Unit (SiLU) in $p_\theta$. Finally, the latent projection layer imposes a bounded latent prior by applying tanh activation after re-parametrization:

$$\mathbf{z} = \tanh(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}) \tag{1}$$

This enforces a known normalized range, which is beneficial for using these latents as conditioning signals.

### *2.1.2 Processing Network*

Overall, the network processes an input audio block $\mathbf{x} \in \mathbb{R}^{T\times 1}$ given a set of latent conditioning vectors $\mathbf{c} = \{\mathbf{c}_i\}_{i=1}^{B}$ - where $B$ is the number of frequency bands and each $\mathbf{c}_i \in \mathbb{R}^{d_c}$ is derived from the respective latent representation $\mathbf{z}$ - and outputs an audio block $\hat{\mathbf{y}}$, along with a state buffer $\mathbf{s}'$ which preserves the last $R-1$ samples corresponding to the receptive field $R$ (see Sec. 3.2), prepended to the input for causal processing:

$$\hat{\mathbf{y}}, \mathbf{s}' = f_\theta(\mathbf{x}, \mathbf{c}, \mathbf{s}). \tag{2}$$

While vanilla TCNs rely on stacks of dilated convolutions [32], *Prism* exploits structurally refined residual blocks, aiming at improving both robustness and efficiency for real-time usage. Each block features a depth-wise convolutional layer with exponentially increasing di-

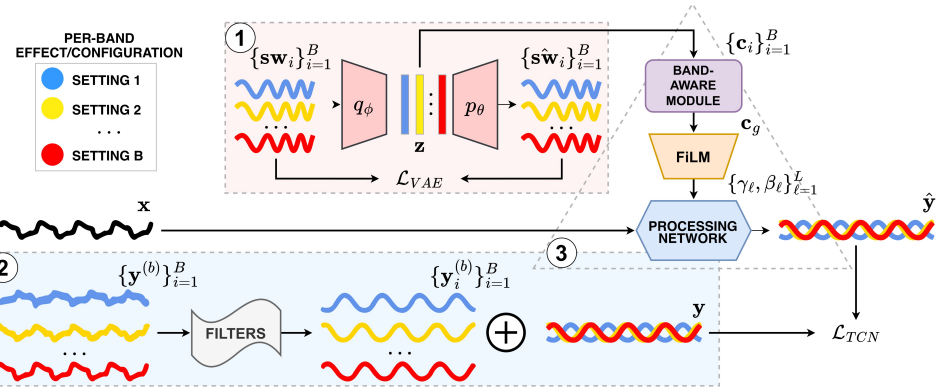



Fig. 1: Summary scheme of the training pipeline, showing: 1. the VAE trained separately on individual per-setting sweeps, extracting latents for conditioning; 2. the pseudo-ground-truth algorithm, which randomly assembles target waveforms for training; and 3. *Prism*: the TCN-based processing network and its conditioning modules. Colored waveforms exemplify the signals decomposed into different bands, whose number $B \in \{2,3,4,6,8\}$ is defined by the chosen model configuration.

lation $d_\ell = 2^{\ell-1}$ and fixed kernel size $k = 3$. In addition, early blocks apply a $1 \times 1$ expansion convolution to increase channel capacity. After the depthwise convolution, group normalization, and SiLU activation replace the typical tanh and $\sigma$ gating of WaveNet-style models [33], yielding smoother gradients and improved stability. FiLM modulation (see 2.1.4) is then applied to inject global conditioning, followed by a Squeeze-and-Excitation (SE) [35] mechanism that adaptively re-weights feature channels based on their global statistics.

The input block first passes through a stem network producing the initial hidden representation $\mathbf{h}_0$. Then, each residual block $\ell$ transforms its input $\mathbf{h}_{\ell-1}$ as

$$\begin{aligned} \mathbf{z}_\ell &= \mathrm{SiLU}(\mathrm{GN}(\mathrm{Conv}_{\mathrm{dw}}(\mathbf{h}_{\ell-1}))), \\ \tilde{\mathbf{z}}_\ell &= \gamma_\ell \odot z_\ell + \beta_\ell, \\ \hat{\mathbf{z}}_\ell &= \mathrm{SE}(\tilde{\mathbf{z}}_\ell) = \tilde{\mathbf{z}}_\ell \odot \sigma\big(\mathrm{W}_2\ \mathrm{PReLU}(\mathrm{W}_1\ \mathrm{GAP}(\tilde{\mathbf{z}}_\ell))\big) \end{aligned} \tag{3}$$

where $\mathrm{Conv}_{\mathrm{dw}}(\cdot)$ denotes a depthwise dilated convolution with pointwise projections before and after, $\mathrm{GN}(\cdot)$ is Group Normalization, and $(\gamma_\ell, \beta_\ell)$ are the layer-specific FiLM parameters derived from the global conditioning vector. The SE module first applies Global Average Pooling $\mathrm{GAP}(\cdot)$ to obtain channel descriptors, then processes them through two learned projections $\mathbf{W}_1$ and $\mathbf{W}_2$ (implemented as $1 \times 1$ convolutions) with PReLU [36] and sigmoid activations, yielding channel-wise gating coefficients.

Finally, the residual output $\mathbf{h}_\ell$ (passed to the subsequent block) and skip connection $\mathbf{s}_\ell$ are computed as

$$\begin{aligned} \mathbf{h}_\ell &= \mathbf{h}_{\ell-1} + 0.5\,\mathrm{Conv}^{1\times1}(\hat{\mathbf{z}}_\ell), \\ \mathbf{s}_\ell &= \mathrm{Conv}^{1\times1}(\hat{\mathbf{z}}_\ell) \end{aligned} \tag{4}$$

where $\mathrm{Conv}^{1\times1}(\cdot)$ denotes a pointwise convolution. The skip connections from all blocks are accumulated and processed through a final output network to produce $\hat{\mathbf{y}}$.

#### 2.1.3 Band-aware conditioning

This module receives the conditioning set $\mathbf{c} = \{\mathbf{c}_i\}_{i=1}^{B}$, encoding information for specific spectral sub-bands. Each $\mathbf{c}_i$ is concatenated with a trainable per-band embedding $\mathbf{e}_i \in \mathbb{R}^{d_e}$ and processed by a shared Multi-Layer Perceptron (MLP) with PReLU activations, ensuring explicit band identity. The resulting band-wise hiddens are then stacked and passed through a final projection layer to produce a single global conditioning vector $\mathbf{c}_{\mathrm{g}} \in \mathbb{R}^{d_{\mathrm{cond}}}$.

#### 2.1.4 FiLM-based modulation

The global conditioning vector $\mathbf{c}_{\mathrm{g}}$ is then mapped to per-layer modulation parameters $\{\gamma_\ell, \beta_\ell\}_{\ell=1}^{L}$ through a standard, lightweight FiLM module, which projects $\mathbf{c}_{\mathrm{g}}$ through a shared two-layer MLP with SiLU activations followed by layer-specific linear heads:

$$[\gamma_\ell, \beta_\ell] = \mathbf{W}_\ell\ \mathrm{SiLU}\big(\mathbf{W}_2\ \mathrm{SiLU}(\mathbf{W}_1\,\mathbf{c}_{\mathrm{g}})\big), \tag{5}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_\ell$ are the learned projection matrices.

### 2.2 Objectives

#### 2.2.1 VAE

The VAE is trained in an unsupervised fashion, whose objective is to maximize the Evidence Lower Bound (ELBO), balancing reconstruction accuracy and latent space regularization:

$$\mathscr{L}_{VAE} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{sw})}\left[\log p_\theta(\mathbf{sw}|\mathbf{z})\right] - D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}|\mathbf{sw}) \| p(\mathbf{z})\right) \tag{6}$$

While the second term is the Kullback-Leibler (KL) divergence, the first one deserves a more detailed explanation. It refers to the reconstruction loss corresponding to the sum of Mean-Squared Error (MSE) loss, as in (7), and Multi-Resolution Short-Time Fourier Transform (MR-STFT) loss [37], computed as the mean of both linear and log-magnitude differences between predicted and target spectrograms over multiple STFT scales $s_i \in W\{1024, 512, 256, 128, 64\}$, as shown in (8).

$$\mathscr{L}_{\mathrm{MSE}} = \|\mathbf{sw} - \mathbf{s\hat{w}}\|_2^2, \tag{7}$$

PREPRINT, Published DOI: https://doi.org/10.17743/jaes.2026.0278

$$\mathscr{L}_{\text{STFT}} = \frac{1}{N}\sum_{i=1}^{N}\left(\|\mathbf{S}_{\hat{\mathbf{sw}}}^{(i)} - \mathbf{S}_{\mathbf{sw}}^{(i)}\|_1 + \|\log(\mathbf{S}_{\hat{\mathbf{sw}}}^{(i)} + \varepsilon) - \log(\mathbf{S}_{\mathbf{sw}}^{(i)} + \varepsilon)\|_1\right) \tag{8}$$

In (8), $\mathbf{S}_{\hat{\mathbf{sw}}}^{(i)} = |\text{STFT}_{s_i}(\hat{\mathbf{sw}})|^2$ and $\mathbf{S}_{\mathbf{sw}}^{(i)} = |\text{STFT}_{s_i}(\mathbf{sw})|^2$ denote the magnitude spectrograms of the predicted and ground-truth signals at scale $s_i$, and $\varepsilon$ is a small constant.

#### 2.2.2 Processing Network

In line with the literature, e.g., [9, 15, 20], the model is trained to minimize a composite loss objective, balancing temporal/spectral accuracy and perceptual consistency by combining three complementary elements: the MR-STFT loss defined above, an Error-to-Signal Ratio (ESR) loss, and a per-band Mean-Squared Error (bMSE) loss. The ESR loss promotes convergence by explicitly accounting for energy variations across training samples, preventing underfitting, especially in low-energy audio blocks:

$$\mathscr{L}_{\text{ESR}} = \frac{\mathbb{E}\left[(\mathbf{y} - \hat{\mathbf{y}})^2\right]}{\mathbb{E}\left[\mathbf{y}^2\right] + \varepsilon}. \tag{9}$$

The predicted and target signals are split into $B$ mel-frequency bands, and a per-band weighted MSE is applied:

$$\mathscr{L}_{\text{bMSE}} = \frac{1}{B}\sum_{b=1}^{B}\lambda_b \cdot \text{MSE}(\hat{\mathbf{y}}_b, \mathbf{y}_b), \tag{10}$$

where $\lambda_b$ is a scaling factor relative to each band; higher frequency bands are given more relevance to better capture fine-grained details. The total *Prism* objective is expressed as the sum of these three components:

$$\mathscr{L}_{\text{TCN}} = \mathscr{L}_{\text{STFT}} + \mathscr{L}_{\text{ESR}} + \mathscr{L}_{\text{bMSE}}. \tag{11}$$

## 3 EXPERIMENTAL SETUP

### 3.1 Dataset

Three well-known nonlinear analog pedals widely used in modern guitar practice were selected for modeling, covering a wide range of timbral characteristics: an overdrive - AnalogMan King of Tone (`KoT`), a distortion - Suhr Riot Reloaded (`RR`), and a Muff-based [18] fuzz - Jam Red Muck (`RM`). As for the overdrive, `KoT` recordings were taken from the *pOD-set* dataset [9], which provides high-quality recordings of individual settings, for a total of 36 gain/tone knobs combinations. The profiling procedure of the remaining two pedals follows the same setup used in [9]: a ∼6-minute input file, containing 4-second sine sweeps along with multisource recordings of various instruments originally presented in [38], is processed through each effect, for all combinations of 6 gain/tone settings. Recordings were taken at 48 kHz, 24-bit, and normalized at -3 dB, compensating audio latency for temporal alignment. The whole dataset contains ∼10 hours of audio data.

### 3.2 Training Pipeline and Hyperparameters

A series of training cycles is conducted, in which separate models are trained with an increasing number of bands, $B \in 2,3,4,6,8$. An additional cycle with $B = 1$ is performed to provide a baseline for validation.

Offline ground-truth data construction procedure is summarized in Algorithm 1. For each input block $\mathbf{x} \in \mathbb{R}^T$, the dataloader retrieves $B$ processed versions of the same segment, each corresponding to a distinct effect configuration. Each waveform is then decomposed into $B$ mel frequency bands with ±5% overlap using a bank of linear-phase, 513-coefficient FIR filters, implemented as a set of 1-D convolutions, producing band-limited signals $\{\mathbf{y}_i^{(b)}\}_{i=1}^{B}$ with minimal phase distortion. A composite pseudo-ground-truth $\mathbf{y}$ is constructed by randomly selecting, for each band index $i$, one of the $B$ effect configurations, then summing them. The conditioning tensor $\mathbf{c}$ is formed accordingly by concatenating the corresponding latent vectors.

**Algorithm 1** Pseudo-Ground-Truth Random Assembly

**Input:** Input block $\mathbf{x} \in \mathbb{R}^T$, filterbank $\mathscr{B}_{\text{mel}}$ with $B$ bands, processed samples $\{\mathbf{y}^{(b)}, \mathbf{c}^{(b)}\}_{b=1}^{B}$
**Output:** Composite pseudo-ground-truth waveform $\mathbf{y}$ and concatenated conditioning tensor $\mathbf{c}$

1. **for** $b = 1,\ldots,B$:
   Decompose processed sample into mel bands:
   $\mathbf{Y}^{(b)} = \mathscr{B}_{\text{mel}}(\mathbf{y}^{(b)}) = \{\mathbf{y}_1^{(b)},\ldots,\mathbf{y}_B^{(b)}\}$
2. **for** $i = 1,\ldots,B$:
   Randomly select $b_i \sim \mathscr{U}\{1,\ldots,B\}$
   Assign band component $\mathbf{y}_i = \mathbf{y}_i^{(b_i)}$
   Assign corresponding latent vector $\mathbf{c}_i = \mathbf{c}^{(b_i)}$
3. Assemble pseudo-ground-truth: $\mathbf{y} = \sum_{i=1}^{B}\mathbf{y}_i$
4. Stack per-band conditioning: $\mathbf{c} = [\mathbf{c}_1^\top;\ldots;\mathbf{c}_B^\top]$

**return** $\mathbf{y}$, $\mathbf{c}$

Finally, the predicted processed output $\hat{\mathbf{y}}$ is also decomposed into sub-bands via the same Mel filterbank, and immediately recomposed before loss computation. While this step may seem redundant, it guarantees spectral consistency between prediction and target and prevents minor artifacts that could arise from mismatched filter responses.

Models are initialized with eight residual blocks, yielding a receptive field $R$ = 511 samples (∼10 ms). Early blocks use multiplicative expansion factors, while later blocks retain the base width of 64 channels. The band conditioning module produces a 64-D global representation $\mathbf{c}_g$ for the FiLM module. All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU, using the AdamW optimizer with an initial learning rate of $1 \times 10^{-3}$ and weight decay of $1 \times 10^{-5}$, a linear decay schedule reducing the learning rate to $1 \times 10^{-6}$, and a batch size of 64. Models are trained for 300 epochs using an 80/20% train/test split, with audio block sizes of 2048 samples at a sampling rate of 48 kHz.

### 3.3 Additional Evaluation and Ablations

For an extensive assessment of the proposed pipeline and to support consistent benchmarking, a structured series of in-depth experiments has been performed. Each configura-

tion involved training the entire pipeline through complete training cycles, covering the following settings:

1. **Individual setting** - [KoT, RR, RM]: models trained individually on each of the three pedals considered;
2. **Pair settings** - [KoT+RR, KoT+RM, RR+RM]: models trained on pairs of pedals;
3. **Monotype setting** - [KoT+DK+JR]: models trained on three pedals, each belonging to the same type of nonlinear effect - overdrives. Here, further data from the *pOD-Set* dataset [9] have been included, namely Tanabe Dumkudo (DK) and Vemuram Jan Ray (JR);
4. **Multitype setting, alt. cond.** - [KoT+RR+RM]: replication of the main experiments using alternative conditioning strategies to evaluate their impact on models' accuracy, specifically: (i.) standard discrete conditioning (effect type, gain, tone) as in e.g., [20, 38], and (ii.) a 4-dimensional latent conditioning scheme.

Performing all aforementioned experiments across every choice of $B$ would be computationally demanding: for this reason, training cycles have been performed only for $B \in \{1,4\}$, which still provides a representative and sufficiently informative set of conditions for evaluation.

## 4 RESULTS

### 4.1 Audio Quality Metrics

Reconstruction metrics for all the multiband models and 1-band baseline are reported in Table 1. Overall, performance is inversely proportional to the number of bands. This reflects the increasing difficulty for the network to learn end-to-end multiband transfer functions. As the pseudo-ground-truth incorporates more frequency bands, the waveform structure becomes more complex and challenging to reconstruct. This challenge is further aggravated when models are conditioned with very different settings, e.g., low-gain overdrive in one band and very high-gain distortion in another, producing a highly intricate target waveform - Fig. 2. Despite this, the models still capture the signal structure, and the degradation is smooth and sufficiently predictable.

| Bands | STFT | ESR | MSE ($\times 10^{-3}$) | MAE ($\times 10^{-2}$) |
|---|---|---|---|---|
| 2 | 0.362 | 0.053 | 0.582 | 1.105 |
| 3 | 0.538 | 0.092 | 1.031 | 1.656 |
| 4 | 0.613 | 0.117 | 1.214 | 1.885 |
| 6 | 0.843 | 0.168 | 1.715 | 2.512 |
| 8 | 0.959 | 0.204 | 2.109 | 2.927 |
| 1 | 0.318 | 0.035 | 0.377 | 0.903 |

Table 1: Evaluation metrics on unseen data across band configurations. The 1-band case is shown as a baseline.

### 4.2 Computational Efficiency

Table 2 reports efficiency metrics, specifically MFLOPs on the non-scripted model graph, and Real-Time Factor (RTF) and Average Forward Time (AFT), computed on an Apple M2 processor, on TorchScript-optimized versions - averaged over 1000 forward passes. Each metric is computed with random mock signals of 512 samples, compatible with the models' receptive fields. In this configuration, all AFTs are confidently below 10 ms, resulting in an acceptable latency. Higher block sizes further improve RTF but increase latency, and are therefore not suitable for real-time audio processing; despite that, for the sake of completeness, performances for 1024, 2048, and 4096 block sizes are still reported in Fig. 3. Reduced variance of performances across band number reflects the single-pass processing approach.

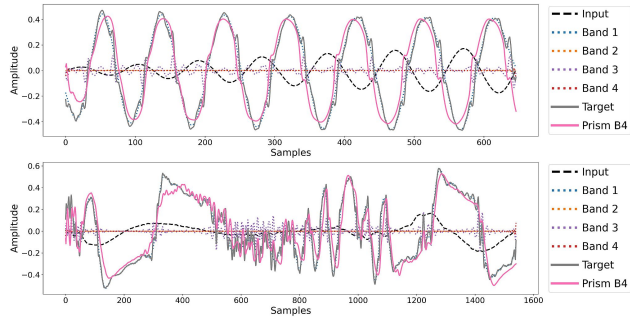



Fig. 2: Two examples of comparison between target waveforms (pseudo-ground-truth) and the predicted outputs of the 4-band model with the following per-band settings: B1 - RR(5,5), B2 - KoT(0,0), B3 - RR(5,5), B4 - RM(1,0). Input and real filtered bands are also shown.

| Bands | Params | MFLOPs | AFT (ms) | RTF |
|---|---|---|---|---|
| 2 | 338,781 | 169.65 | 7.289 | 0.683 |
| 3 | 342,885 | 169.65 | 7.224 | 0.677 |
| 4 | 346,989 | 169.66 | 7.280 | 0.682 |
| 6 | 355,197 | 169.68 | 7.415 | 0.695 |
| 8 | 363,405 | 169.70 | 7.467 | 0.700 |
| 1 | 334,677 | 169.64 | 7.172 | 0.672 |

Table 2: Comparison of model performance across band configurations. The 1-band case constitutes a baseline.

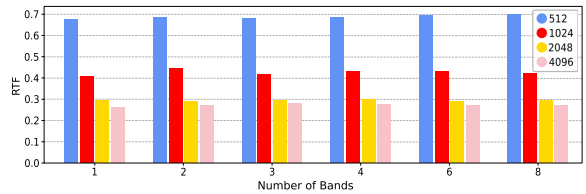



Fig. 3: RTF comparison depending on audio block size.

### 4.3 Models Assessment and Ablation

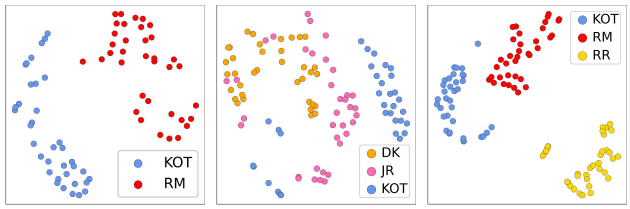
Table 3 reports reconstruction metrics for individual, pair, and monotype settings, comparing 1-band and 4-band configurations. First, consistently with the results in Table 1, increasing the number of bands always leads to higher reconstruction errors across all metrics. Moreover, results for individual effects suggest that overdrive circuits are easier to model. Although this may depend on the specific de-

PREPRINT, Published DOI: https://doi.org/10.17743/jaes.2026.0278

| Setting | Individual | | | | | | Pairs | | | | | | Monotype | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | KoT | | RR | | RM | | KoT+RR | | KoT+RM | | RR+RM | | KoT+DK+JR | |
| **Bands** | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 4 |
| **STFT** | 0.021 | 0.095 | 0.514 | 0.726 | 0.283 | 0.434 | 0.307 | 0.586 | 0.184 | 0.355 | 0.457 | 0.778 | 0.033 | 0.154 |
| **ESR** | 0.010 | 0.059 | 0.026 | 0.077 | 0.035 | 0.072 | 0.027 | 0.114 | 0.040 | 0.086 | 0.034 | 0.112 | 0.011 | 0.060 |
| **MSE** ($\times 10^{-3}$) | 0.016 | 0.093 | 0.600 | 1.852 | 0.280 | 0.618 | 0.352 | 1.426 | 0.197 | 0.418 | 0.525 | 1.835 | 0.028 | 0.192 |
| **MAE** ($\times 10^{-2}$) | 0.066 | 0.346 | 1.222 | 2.646 | 0.698 | 1.376 | 0.856 | 2.037 | 0.444 | 1.022 | 1.063 | 2.553 | 0.143 | 0.607 |

Table 3: Results of the additional evaluation on individual, pair, and monotype settings across 1 and 4-band configurations.

vice, the monotype results confirm this point: even when trained on three different devices (KoT+DK+JR), they still outperform models trained on a single Fuzz (RM) or Distortion (RR), with the latter yielding the worst performance. Similar patterns can also be identified in the pair settings, with KoT contributing to increasing evaluation metrics and RR decreasing them. This variability is also already reflected in the latent representation produced by the VAE - Fig. 4. Indeed, latent plots relative to the pair and multitype settings show well-defined clusters, while the monotype distribution appears overall more uniform - arguably suggesting a less complex structure for the model to learn. Cluster characteristics were found to be robust to t-SNE hyperparameter tuning.




(a) (b) (c)

Fig. 4: t-SNE on VAE latents for different settings: (a) Pairs; (b) Monotype; (c) Multitype.

Differences between the models also emerge in the frequency-response analysis, where processed sinusoidal sweeps (g=5, t=5) from the dataset have been reused as input for individually assessing the three pedals over three model settings: 1) individual; 2) multitype single-band; and 3) multitype 8-band - Fig. 5.

For the individual settings, the estimated responses closely match the ground-truth curves across all effects. In contrast, the multitype models - both single-band and multiband - exhibit more distinctive behaviors.

Specifically, for KoT, both multitype models begin to deviate from the real response at ∼150 Hz, remaining consistently lower throughout the mid-range; such deviation is more pronounced in the 8-band model. However, at higher frequencies, both multitype versions approach the ground truth more closely than the individually-trained model.

For RM, both multitype models produce lower amplitudes in the low and high frequency regions and slightly higher amplitudes in the mid-range. Again, the 8-band variant expands this trend.

In the case of RR, the single-band and 8-band multitype models show nearly identical behavior in the low-frequency region - both slightly lower than the ground-truth. Interestingly, above roughly 200 Hz, the 8-band model starts to align almost perfectly with the real response, whereas the single-band version continues to diverge slightly. This is in contrast with the evaluations in both Table 1 - where the 8-band model performs worse on average - and Table 3 - where individual and pair settings involving RR also yield worse results.

Finally, Table 4 reports the ablation results on the conditioning signal. In the case of the single-band multitype setting, all the conditioning modalities perform similarly, with the discrete one even slightly better overall. However, when the number of bands increases, latent conditioning outperforms the discrete one in all metrics, signifying that the conditioning modules benefit from structured representation proportionally as intra-band complexity grows.

| Bands | Cond. | STFT | ESR | MSE ($\times 10^{-3}$) | MAE ($\times 10^{-2}$) |
|---|---|---|---|---|---|
| 1 | Discrete | 0.320 | 0.031 | 0.327 | 0.746 |
| 1 | 4 | 0.327 | 0.036 | 0.403 | 0.933 |
| 1 | 8 | 0.318 | 0.035 | 0.377 | 0.903 |
| 4 | Discrete | 0.696 | 0.126 | 1.397 | 2.087 |
| 4 | 4 | 0.619 | 0.120 | 1.214 | 1.924 |
| 4 | 8 | 0.613 | 0.116 | 1.206 | 1.885 |

Table 4: Ablation study for different conditioning strategies on the KoT+RR+RM setting.

## 5 INFERENCE AND INTERFACE

The 8-band version of the final multitype model was complemented with a graphical user interface, granting quick access to the choice of effect type, gain, and tone for each frequency band. Fig. 6 presents the user interface while Fig. 7 details the real-time inference process. Gain, tone, and effect parameters for each band are used to retrieve the latent vectors previously extracted from the whole dataset. The assembled conditioning vector $\mathbf{c} = [\mathbf{c}_1^\top; \ldots; \mathbf{c}_8^\top]$ - relative to all bands - is then fed to the processing network.

The interface was developed with the JUCE C++ framework, and compiled as a standalone executable and audio plugin. The current version sends open sound control messages to a Python runtime that processes the sound. Future

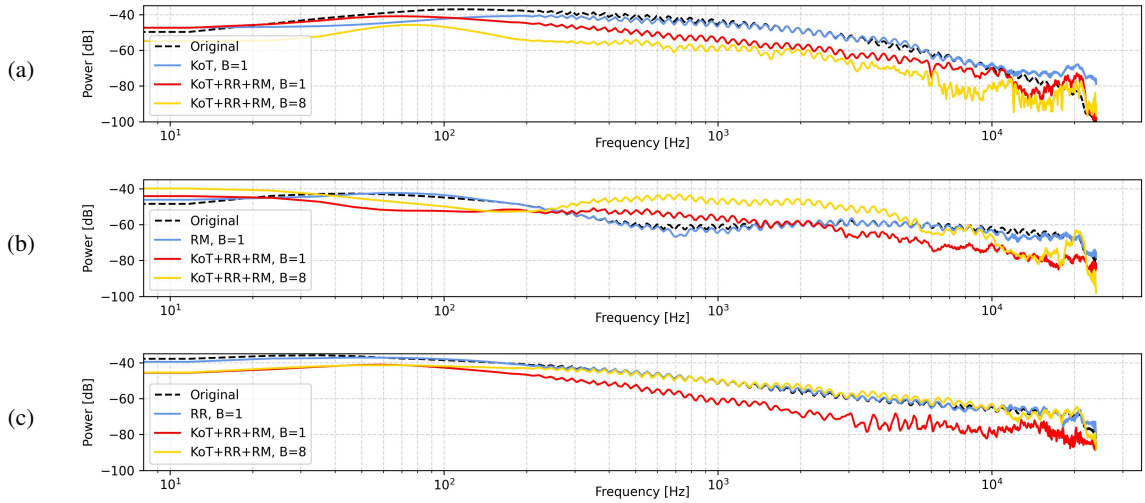



Fig. 5: Frequency responses on sine sweeps of the ground-truth pedal output and three models for (a) `KoT`, (b) `RM`, and (c) `RR`. Each panel compares the real processed signal (black) with the individual (blue) and 1-band multitype (red) baselines, and the more complex multitype 8-band model (gold). All plots refer to the same `(5,5)` configuration.

work will focus on integrating the model inference directly into the plugin.


Fig. 6: The graphical interface for real-time usage.

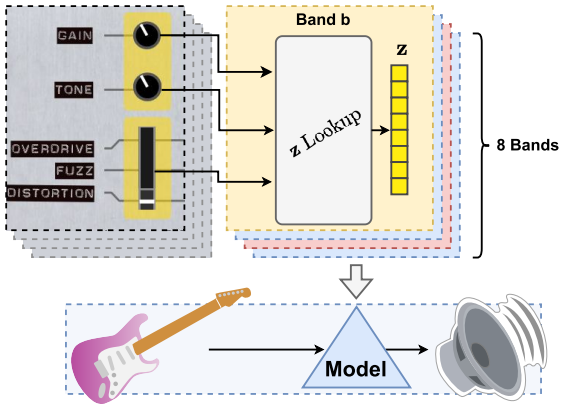



Fig. 7: Real-time inference process. Per-band settings are used to retrieve conditioning vectors $\mathbf{z}$ encoded by the VAE. Latent vectors are fed to the conditioning input of the modeling network along with real-time audio blocks.

## 6 DISCUSSIONS

Table 5 provides a comprehensive performance comparison of several state-of-the-art methods for real-time audio processing against two versions - 2 and 8 bands - of *Prism*, namely the less and more complex settings, respectively. All of these works present conventional single-band models trained on single devices.

Despite the larger complexity both in terms of the number of devices modeled simultaneously and explicit multiband conditioning, *Prism* achieves optimal audio quality that is in line with, and in some settings superior to, those reported in prior works. For instance, considering the 2-band model, only Comunità *et al.* [15, 39] report marginally lower aggregated errors on the metrics available on their models. However, [15] targets substantially narrower modeling tasks, where a single LSTM network is trained on a Fuzz device conditioned solely on the gain setting. On the contrary, [39] trained a Gated Convolutional Network (GCN) on a "custom fuzz" circuit conditioned on multiple gain+attack+release settings, suggesting a potential hybrid effect behavior; while this may constitute a closer approach to *Prism*, the authors do not provide a detailed analysis of their circuit to assess its internal interdependency. Dal Rí *et al.* [9] recently proposed a system to morph between multiple overdrive settings using a tiny GRU network, representing a related method to creatively unify multiple devices in a single model; however, it operates on the full spectrum and is limited to a single type of effect. In addition, distortion and fuzz being more difficult to emulate aligns with the findings in [15].

All other surveyed works show equal or larger errors on all provided metrics with respect to the 2-band setting, with the 8-band one still achieving competitive results.

It is worth noting that several works in the literature report substantially lower parameter counts for their models, up to one or even two orders of magnitude smaller than the

PREPRINT, Published DOI: https://doi.org/10.17743/jaes.2026.0278

| Authors | Effect Type | Architecture | Conditioning | SR (kHz) | STFT | ESR | MSE ($\times 10^{-3}$) | MAE ($\times 10^{-2}$) |
|---|---|---|---|---|---|---|---|---|
| Comunitá *et al.* [15] | OD (Amp) | S4-TVF-L-16 | G+T$_{b,m,t}$ | 48 | 0.988 | *n.d.* | *n.d.* | 2.570 |
| Comunitá *et al.* [15] | F | LSTM-96 | G | 48 | 0.152 | *n.d.* | *n.d.* | 0.400 |
| Comunitá *et al.* [39] | F | GCNTF-3 | G+A+R | 44.1 | 0.279 | *n.d.* | *n.d.* | 1.100 |
| Dal Rí *et al.* [9] | OD (multi) | GRU | G+T (emb) | 48 | 0.814 | *n.d.* | 0.836 | *n.d.* |
| Simionato and Fasciani [20] | OD | LSTM | G+T | 48 | 0.750 | 0.113 | 1.710 | *n.d.* |
| Simionato and Fasciani [40] | OD | FiLM-GCU | G+T | 48 | 0.406 | 0.176 | 0.292 | *n.d.* |
| Vanhatalo *et al.* [5] | D (Amp) | LSTM-32 | G | 44.1 | 0.595 | 0.024 | 4.000 | 3.780 |
| ***Prism* (2 bands)** | OD+D+F | TCN | G+T (emb) | 48 | 0.362 | 0.053 | 0.582 | 1.105 |
| ***Prism* (8 bands)** | OD+D+F | TCN | G+T (emb) | 48 | 0.959 | 0.204 | 2.109 | 2.927 |

Table 5: Comparison of different real-time methods from the literature against the best (simpler, B=2) and worse (more complex, B=8) multiband/multitype *Prism* models.

one in *Prism* (e.g., [9, 15, 38]). However, given the complexity of the task and the fact that the proposed models still operate in real-time on a standard CPU, increasing the network capacity has been an intentional design choice. In the current implementation, the core processing network contains 181,258 trainable parameters; the remaining ones stem from the conditioning modules, whose size slightly increases proportionally with the number of bands.

Therefore, beyond faithful emulation of three different types of nonlinear effects simultaneously, *Prism* constitutes a novel approach in the field, and its multiband control capabilities allow for a series of creative applications that are not possible with standard VA modeling. Fig. 8 highlights band-specific behaviors: low-gain settings create spectral gaps where certain bands contribute minimally, while higher-gain settings produce localized amplitude increases.

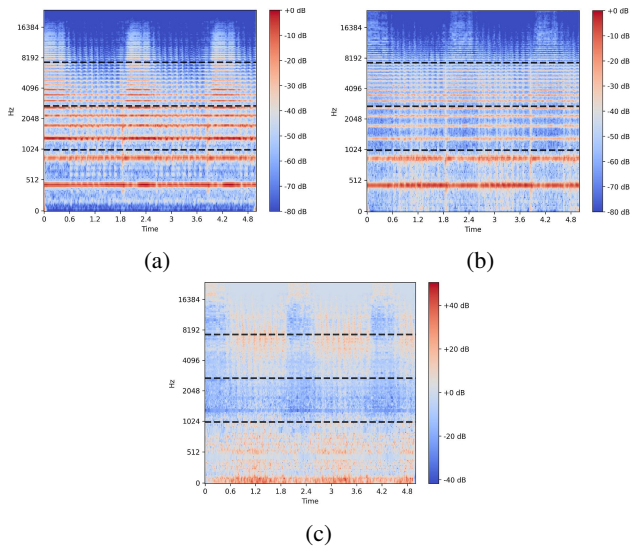



Fig. 8: Log-mel spectrograms of (a) the input signal, (b) the model output, and (c) the difference between the two. The audio is processed using the 4-band model with the following per-band settings: B1 - RR (5,5), B2 - KoT (0,0), B3 - RR (5,5), B4 - RM (1,0).

## 7 CONCLUSIONS

This paper presented *Prism*, a novel approach to neural audio modeling that enables fine-grained control on multiband and multieffect settings simultaneously. Bypassing explicit parallel multiband processing and operating directly on raw waveforms in a single forward pass, *Prism* allows for fast and intuitive audio manipulations and enables straightforward and creative hybrid effects design. Future work will investigate tighter integration of the model into Digital Audio Workstation environments, perceptual evaluations with expert end-users, and potential extensions to other classes of nonlinear processors.

## 8 REFERENCES

[1] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," in U. Zölzer (Ed.), *DAFX: Digital Audio Effects*, pp. 473–522 (Wiley Online Library, 2011), doi: 10.1002/9781119991298.

[2] R. Giampiccolo, S. C. Gafencu, and A. Bernardini, "Explicit Modeling of Audio Circuits With Multiple Nonlinearities for Virtual Analog Applications," *IEEE Open Journal of Signal Processing*, vol. 6, pp. 156–164 (2025), doi:10.1109/OJSP.2025.3528334.

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

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

[5] T. Vanhatalo, P. Legrand, M. Desainte-Catherine, P. Hanna, A. Brusco, G. Pille, *et al.*, "A Review of Neural Network-based Emulation of Guitar Amplifiers," *Applied Sciences*, vol. 12, no. 12, p. 5894 (2022), doi:10.3390/app12125894.

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

[7] D. J. Gillespie and D. P. Ellis, "Modeling nonlinear circuits with linearized dynamical models via kernel regression," presented at the *IEEE Worksh. on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4 (2013), doi:10.1109/WASPAA.2013.6701830.

[8] S. Orcioni, A. Terenzi, S. Cecchi, F. Piazza, and A. Carini, "Identification of Volterra models of tube audio devices using multiple-variance method," *Journal of the Audio Engineering Society*, vol. 66, no. 10, pp. 823–838 (2018), doi:10.17743/jaes.2018.0046.

[9] F. A. Dal Rí, D. Stefani, L. Turchet, and N. Conci, "Morphdrive: Latent Conditioning For Cross-Circuit Effect Modeling And A Parametric Audio Dataset Of Analog Overdrive Pedals," in *Proc. of 28th Int. Conf. on Digital Audio Effects (DAFx25)*, pp. 23–30, 2025 Sept).

[10] R. Simionato and S. Fasciani, "Hybrid Neural Audio Effects," in *Proc. of the Sound and Music Computing (SMC) Conf.*, 2024).

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

[12] S. Prince and K. Shankar Kumar, "Survey on effective audio mastering," in *Proc. of Int. Conf. on Future Generation Communication and Networking*, pp. 293–301, 2012), doi:10.1007/978-3-642-35594-3_41.

[13] P. Fernandez-Cid and J. Casajus-Quiros, "Multiband approach to digital audio FX," in *Proc. of IEEE Int. Conf. on Multimedia and Expo (ICME)*, pp. 1747–1750, 2000).

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

[15] M. Comunità, C. J. Steinmetz, and J. Reiss, "Differentiable black-box and gray-box modeling of nonlinear audio effects," *Frontiers in Signal Processing*, vol. 5, p. 1580395 (2025), doi:10.3389/frsip.2025.1580395.

[16] D. Südholt, A. Wright, C. Erkut, and V. Välimäki, "Pruning deep neural network models of guitar distortion effects," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 256–264 (2022), doi:10.1109/TASLP.2022.3223257.

[17] A. Carson, A. Wright, and S. Bilbao, "Anti-aliasing of neural distortion effects via model fine tuning," in *Proc. of 28th Int. Conf. on Digital Audio Effects (DAFx25)*, pp. 15–22, 2025).

[18] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Applied Sciences*, vol. 10, no. 3, p. 766 (2020).

[19] L. Juvela, E.-P. Damskägg, A. Peussa, J. Mäkinen, T. Sherson, S. I. Mimilakis, *et al.*, "End-to-end amp modeling: from data to controllable guitar amplifier models," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023), doi:10.1109/ICASSP49357.2023.10094769.

[20] R. Simionato and S. Fasciani, "Comparative study of state-based neural networks for virtual analog audio effects modeling," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2025, no. 1, p. 30 (2025).

[21] R. Simionato and S. Fasciani, "Modeling time-variant responses of optical compressors with selective state space models," *Journal of the Audio Engineering Society*, vol. 73, pp. 144–165 (2025).

[22] E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time modeling of audio distortion circuits with deep learning," in *Sound and Music Computing Conf. (SMC)*, pp. 332–339, 2019).

[23] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time modeling of analog dynamic range compression," in *152nd AES Convention*, 2021).

[24] V. Huhtala, L. Juvela, and S. J. Schlecht, "KLANN: Linearising long-term dynamics in nonlinear audio effects using Koopman networks," *IEEE Signal Processing Letters*, vol. 31, pp. 1169–1173 (2024), doi:10.1109/LSP.2024.3389465.

[25] C. J. Steinmetz and J. D. Reiss, "Randomized overdrive neural networks," presented at the *4th Worksh. on Machine Learnign for Creativity and Design, NeurIPS* (2020).

[26] D. Kim, S. Park, D. K. Han, and H. Ko, "Multiband CNN Architecture Using Adaptive Frequency Filter for Acoustic Event Classification," *Applied Acoustics*, vol. 172, p. 107579 (2021), doi:10.1016/j.apacoust.2020.107579.

[27] J. Zheng, M. Cao, and C. Zhang, "Chinese Instrument Music Source Separation with Frequency-attentive Multi-band Neural Networks," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2025, no. 1, p. 36 (2025), doi:10.1186/s13636-025-00423-4.

[28] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, "Multi-band MelGAN: Faster Waveform Generation for High-quality Text-to-speech," presented at the *IEEE Spoken Language Technology Worksh. (SLT)*, pp. 492–498 (2021), doi:10.1109/SLT48900.2021.9383551.

[29] S. I. Mimilakis, K. Drossos, T. Virtanen, and G. Schuller, "Deep neural networks for dynamic range compression in mastering applications," in *140th AES Convention*, 2016).

[30] H. S. Ioniţă, V. Popa, and B. Moroşanu, "Multiband Dynamics Compressor with Automatic Parameter Learning," in *Proc. of 15th Int. Conf. on Communications (COMM)*, pp. 1–4, 2024), doi:10.1109/COMM62355.2024.10741473.

[31] P. Sarkar and P. Lindborg, "Neural-Driven Multi-Band Processing for Automatic Equalization and Style Transfer," in *Proc. of 28th Int. Conf. on Digital Audio Effects (DAFx25)*, pp. 382–389, 2025).

[32] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *ICLR* (2018).

[33] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, *et al.*, "WaveNet: A Generative Model for Raw Audio," in *9th ISCA Worksh. on Speech Synthesis (SSW 9)*, p. 125, 2016), doi:10.48550/arXiv.1609.03499.

PREPRINT, Published DOI: https://doi.org/10.17743/jaes.2026.0278

[34] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual Reasoning with a General Conditioning Layer," in *32th AAAI Conf. on Artificial Intelligence*, AAAI'18 (AAAI Press, 2018), doi:10.1609/aaai.v32i1.11671.

[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018), doi:10.1109/CVPR.2018.00745.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 1026–1034, 2015), doi:10.1109/ICCV.2015.123.

[37] 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 *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6199–6203, 2020), doi:10.1109/ICASSP40776.2020.9053795.

[38] Y.-T. Yeh, W.-Y. Hsiao, and Y.-H. Yang, "Hyper Recurrent Neural Network: Condition Mechanisms for Black-box Audio Effect Modeling," in *Proc. of 27th Int. Conf. on Digital Audio Effects (DAFx24)*, pp. 97–104, 2024).

[39] M. Comunità, C. J. Steinmetz, H. Phan, and J. D. Reiss, "Modelling black-box audio effects with time-varying feature modulation," in *Proc. of IEEE Int Conf on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023), doi:10.1109/ICASSP49357.2023.10097173.

[40] R. Simionato and S. Fasciani, "Conditioning methods for neural audio effects," in *Proc. of the Sound and Music Computing (SMC) Conf.*, 2024).

## THE AUTHORS


Francesco Ardan Dal Rí


Domenico Stefani


Luca Turchet


Nicola Conci

Francesco Ardan Dal Rí is a musician and Ph.D student at the Dept. of Information Engineering and Computer Science of the University of Trento, Italy. He holds a Master's Degree in Electronic Music from the Conservatory F.A. Bonporti of Trento. His current research interests span from generative audio to neural synthesis.

●

Domenico Stefani is an assistant professor at the Dept. of Information Engineering and Computer Science of the University of Trento, Italy. He received a master's degree in Computer Science from the University of Trento, Italy, and a Ph.D. degree from the same institution in 2024. His research interests include musical audio processing, music information retrieval, and embedded systems.

●

Luca Turchet is an Associate Professor at the Dept. of Information Engineering and Computer Science at the University of Trento, Italy. He holds master's degrees in Computer Science (University of Verona), classical guitar and composition (Music Conservatory of Verona), and electronic music (Royal College of Music, Stockholm), and a Ph.D. in Media Technology from Aalborg University Copenhagen. He is Chair of the IEEE Emerging Technology Initiative on the Internet of Sounds, founding president of the Internet of Sounds Research Network, and a former Associate Editor of the Journal of the Audio Engineering Society.

●

Nicola Conci is Associate Professor with the Dept. of Information Engineering and Computer Science, University of Trento. His current research interests are in the field of multimedia signal processing, in particular, computer vision applications for human behavior understanding, at the intersection between vision, graphics, and machine learning, authoring more than 160 publications in international peer-reviewed conferences and journals.