

SPRING REVERB EMULATION WITH HYBRID GATED CONVOLUTIONAL NETWORKS AND STATE SPACE MODELS

Jonas Janser¹, Matthias Wess^{1,2}, Dominik Dallinger^{1,2},
Matthias Bittner^{1,2}, Daniel Schnöll^{1,2}, Axel Jantsch^{1,2}

¹ Institute of Computer Technology, TU Wien, Austria

² Christian Doppler Laboratory for Embedded Machine Learning, TU Wien, Austria

ABSTRACT

Modeling analog audio effects like spring reverbs is a long-standing challenge due to their complex, nonlinear behaviors, such as amplitude-dependent transients and long, dispersive reverberant tails. While deep learning has shown promising results, existing approaches often struggle to capture these characteristics simultaneously. In this paper, we propose **GCN-SSM**, a novel hybrid model architecture that combines a Gated Convolutional Network (GCN) with a State Space Model (SSM), leveraging sequential blocks of interleaved feedforward and recurrent layers. We evaluate the performance with spectral and time-domain losses, and a MUSHRA listening test. Our results show that both components are critical for achieving state-of-the-art perceptual quality. The **GCN-SSM** consistently outperforms the non-hybrid GCN across all metrics. With only 125.7k parameters and inference requiring 5.9 GFLOP for 1 second of audio at 44.1 kHz, the **GCN-SSM** is theoretically suitable for deployment on modern CPUs.

Index Terms— audio effects, spring reverb, deep learning, state space models, virtual analog modeling

1. INTRODUCTION

Spring reverberation units are known for their characterful sound, but their electromechanical nature introduces complex behaviors that are challenging to model numerically. These include dispersive attack transients (*splash* effect), input-dependent decay, and nonlinear electromechanical effects. Physical modeling techniques attempt to simulate the system mechanics directly, but accurately modeling the helical structure and nonlinear driver requires sophisticated domain-knowledge and is computationally expensive [1, 2, 3]. Alternatively, Linear Impulse Response (IR) convolution [4] provides an efficient method to capture linear characteristics, but fails to reproduce the dynamic nonlinearities inherent in analog hardware. Recent deep learning approaches offer a promising path for black-box modeling of this particular audio effect: Convolutional Neural Networks (CNNs) with dilated convolutions like WaveNet and Gated Convolutional

Networks (GCNs) function as efficient, learnable feedforward models [5]. Their large receptive fields allow them to learn the overall structure and the long decay envelope of a reverb’s linear impulse response. Their application to diverse audio effects has shown promising results [6, 7, 8].

Earlier recurrent approaches, such as Long Short-Term Memories (LSTMs) [9], maintain a recursive state, allowing them to capture long-range dependencies. More recently, State Space Models (SSMs) have emerged as a more scalable and stable alternative [10, 11, 12] for modeling systems with long-term memory. Their suitability for audio processing was demonstrated in tasks like dynamic range compression [13, 14], where they outperform LSTMs, as well as piano synthesis [15] and general waveform synthesis [16], yet their application to reverberation remains unexplored.

Both paradigms exhibit limitations when used in isolation: Feedforward structures can struggle to maintain phase fidelity [17]. SSMs, while suited for linear IR modeling, fail to reproduce the realistic reverberant tail and introduce ringing artifacts in our experiments.

In this paper, we adopt a hybrid model of interleaved GCN and SSM layers, showing that this design substantially improves phase accuracy and overall perceptual quality. This work makes the following key contributions:

Hybrid Model Architectures: We compare and systematically evaluate hybrid deep learning architectures that combine dilated convolutions with SSM layers for high-fidelity spring reverb emulation.

Real-World Dataset: We introduce a dataset for spring reverb modeling, captured from an Electro-Voice EVT 4500 unit, featuring deliberate offset transients to ensure a robust evaluation of the unique characteristics of a spring’s impulse response.

2. RELATED WORK

Deep learning provides a data-driven, task-agnostic framework for black-box modeling of audio effects, enabling complex input-output mappings. Recent work has begun to apply deep learning paradigms specifically to spring reverbera-

tion. Early work by Ramirez et al. [18] proposed an architecture with adaptive convolutional front- and back-ends, and a LSTM operating in the latent space. However, this model was trained on short 2-second clips at a 16 kHz sample rate, restricting its ability to capture the full bandwidth and long temporal dynamics of analog hardware. More recently, Papaleo et al. [5] systematically compared several end-to-end architectures on a 48 kHz dataset limited to 5-second guitar samples generated by a digital spring reverb emulation [19]. Their findings established the GCN as a strong baseline and confirmed that standard recurrent models such as LSTMs are not well suited for this task. Importantly, their work highlighted two open challenges: (i) the lack of perceptual evaluation, as objective metrics may not align with human judgments, and (ii) the absence of specialized datasets derived from real analog hardware.

Our work directly addresses these gaps. Preliminary experiments showed that standalone SSMs produce audible artifacts when applied to this task. We therefore propose a *hybrid* architecture for the end-to-end training that interleaves SSM layers with a GCN backbone. We introduce a high-resolution 44.1 kHz dataset of real spring reverb recordings for training and evaluation, curated to highlight challenging nonlinearities. Using the GCN as our baseline, we validate the proposed GCN-SSM with both objective metrics and a formal MUSHRA listening test, demonstrating its perceptual advantage.

3. METHODOLOGY

This study investigates deep learning architectures for spring reverb emulation, systematically evaluating the impact of combining feedforward and recurrent components. It consists of a custom dataset¹, a set of models, and a defined training procedure. The overall process is summarized in Fig. 1, and the model architectures are detailed in Fig. 2.

3.1. Dataset and Preparation

Existing public datasets for spring reverb emulation are limited in sample rate and timbral diversity. To create a more challenging and realistic benchmark, we constructed a new dataset totaling over 57 minutes of audio. The data were generated by processing license-free mono audio clips through a hardware Electro-Voice EVT 4500 spring reverb unit. A key feature of this dataset is the deliberate inclusion of the system’s characteristic nonlinear offset transient. To consistently provoke this behavior, all 4-second input signals were truncated at the 1.5-second mark, while the full 4-second reverberant tail was recorded. To compensate for noise, a de-noiser was used to remove a learned profile from the recording chain and hardware effect. Another critical step was compensating

¹The dataset, models, and supplementary materials are available at: <https://Kfseekltsch.github.io/spring-ssm/>

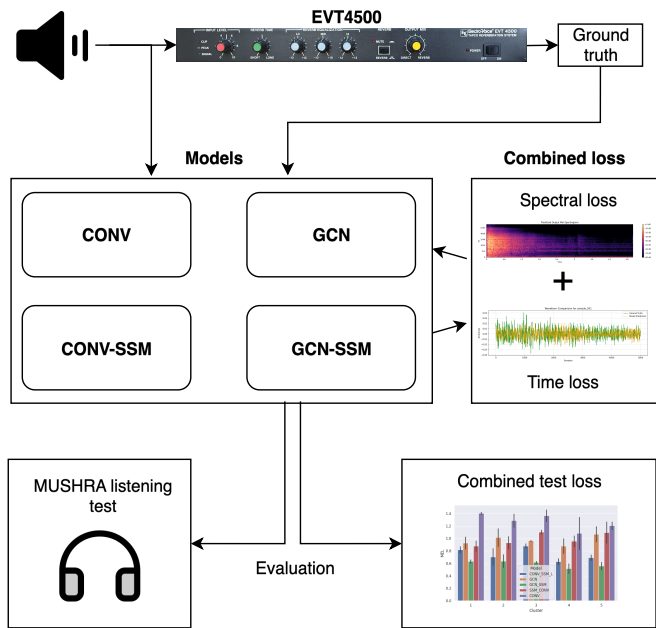


Fig. 1. Overview of the research pipeline: recordings from an EVT4500 spring reverb are used to train the models, evaluated with objective metrics and a MUSHRA listening test.

for the system’s latency. A transient-based cross-correlation analysis between input and output revealed significant signal dispersion, a known characteristic of spring reverbs. Because a single, sample-accurate global alignment is physically impossible, visual alignment was performed on the concatenated audio files. This preserves the small initial dead time between the dry input and the onset of the wet signal, a key perceptual characteristic of the modeled system and its inherent latency. All audio files were stored in mono at 44.1 kHz in 24-bit PCM format.

3.2. Model Architectures

The goal is to evaluate two model setups: (i) a block-based stack of dilated 1D-convolutions followed by a stack of recurrent layers, and (ii) an interleaved design that leverages the gating mechanism and receptive field of GCNs while introducing interleaved SSMs for phase refinement. To systematically assess the contribution of recurrent and feedforward components, we evaluate four architectures:

CONV: A pure feedforward model of 15 dilated convolutional layers ($\text{kernel size } (k_s)=12$, $\text{channels } (ch)=8$), capturing the full reverb tail, followed by a single \tanh . The resulting signals are processed by a projection head (Multi-Layer Perceptron (MLP)) and a final \tanh to produce single-channel output.

CONV-SSM: A sequential feedforward/recurrent model. It uses the same convolutional structure as **CONV** and its activation, followed by 6 SSM layers ($\text{state size}=16$). The

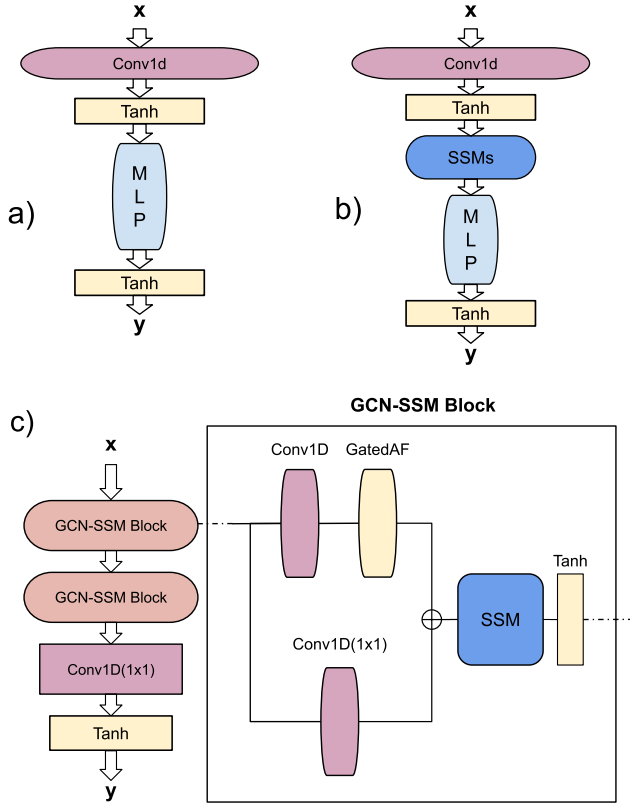


Fig. 2. Model architectures: a) CONV, b) CONV-SSM, c) GCN-SSM. The GCN corresponds to the same design without the interleaved state space layer (SSM) and tanh activation.

projection head matches the **CONV** model.

GCN: A more advanced dilated CNN using the architecture from [20]. It consists of 11 blocks ($k_s=87$, $ch=8$), with gated activations and residual connections. The output head is a dense layer projecting internal channels to a single channel, followed by a *tanh* activation. The receptive field spans 4 seconds.

GCN-SSM: Our main proposal with an interleaved topology. It uses the same GCN structure, but inserts an SSM layer (*state size*=24) with a subsequent *tanh* activation after each GCN block, enabling interaction between feedforward and recurrent components.

For a fair comparison, we include an optimized GCN (**GCN-O**) with $k_s=25$, $ch=16$ based on a hyperparameter sweep. This ensures that observed gains stem from the architecture, not baseline weakness. Similarly we train an optimized GCN-SSM (**GCN-SSM-O**) with *state size*=12 to limit the number of parameters. The resulting parameter counts for all models can be found in Table 2.

3.3. Training

For a mostly deterministic signal-to-signal task, a multi-scale spectral loss balances perceptual relevance and efficiency. We employ a composite loss combining a time-domain Mean Absolute Error (L1) with frequency-domain losses (Multi-Resolution STFT (MRSTFT), Mel-Scaled Multi-Resolution STFT (Mel-MRSTFT)):

$$\mathcal{L} = \mathcal{L}_{L1} + \alpha \mathcal{L}_{\text{Mel-MRSTFT}} + (1 - \alpha) \mathcal{L}_{\text{MRSTFT}}. \quad (1)$$

For the spectral losses, we use the implementations within the *auraloss* library [21], and a weighting of $\alpha = 0.5$.

All models were trained for up to 200 epochs on a single NVIDIA A100 GPU, batch size 6, requiring under 7 hours each. Data were split 70/20/10, while test samples were reserved for evaluation and previews. We used AdamW with a base learning rate of $1e-3$. For models with SSMs, we used parameter groups with different learning rates: core SSM parameters were trained at 10× smaller values than the remaining layers to maintain stability. A *ReduceLROnPlateau* scheduler (patience 15) controlled the learning rate, following a linear 50-epoch warm-up.

4. RESULTS

We evaluate the four models on the test set using both objective metrics and a formal subjective listening test to provide a comprehensive performance assessment.

4.1. Evaluation Setup

Objective Metric: We report five key objective metrics, averaged across the test set: L1, spectral losses (MRSTFT, Mel-MRSTFT), Error-to-Signal Ratio (ESR), and a Magnitude-Weighted Phase Error (Phase) to quantify phase inaccuracies.

Subjective Listening Test: To assess perceptual quality, we conducted a formal MUSHRA (Multiple Stimuli with Hidden Reference and Anchor) test [22]. Fifteen participants rated each model’s similarity to the ground-truth reference on a scale of 0–100 via a web interface. The test included the ground truth as a hidden reference and a single low-pass filtered anchor (3.5 kHz) to ensure reliability.

The overall performance is summarized in Table 1. Our main proposal, the interleaved **GCN-SSM**, achieves the best scores across all objective metrics and obtains the highest MUSHRA score, indicating superior perceptual quality. The GCN baseline significantly outperforms the simpler CONV baseline, confirming the benefit of residual connections and gated activations.

A key finding is that in both cases, the addition of SSM layers (**CONV**→**CONV-SSM** and **GCN**→**GCN-SSM**) leads to substantial improvements in objective metrics. This shows that the refinement from SSMs is critical. MUSHRA results confirm the trend: differences are smaller but still favor SSM-based architectures. Results for the **GCN-SSM-O**, using the

Model	L1 ↓	MRSTFT ↓	Mel ↓	ESR(dB) ↓	Phase ↓	MUSHRA(%) ↑
CONV	0.0049	2.276	2.189	29.49	0.947	56.5
CONV-SSM	0.0055	1.321	1.444	5.30	0.994	63.6
GCN	0.0035	0.682	0.819	0.70	0.626	70.3
GCN-O	0.0040	0.640	0.793	1.01	0.651	72.5
GCN-SSM	0.0016	0.412	0.534	0.13	0.279	79.0
GCN-SSM-O	0.0081	0.438	0.562	2.81	2.001	79.2
reference	-	-	-	-	-	88.6
anchor	-	-	-	-	-	57.0

Table 1. Objective and subjective metrics on the test set. The GCN-SSM achieves the best objective scores, while the optimized version yields the highest MUSHRA score. The ground-truth reference achieves 88.6%, and the low-pass anchor 57%, confirming test reliability.

architecture parameters from the **GCN-O**, reveal an interesting observation: high ESR and phase errors are not reflected in the MUSHRA score. Inspection of the waveform shows a tendency to produce an inverted signal, which remained imperceptible to listeners. This inversion could potentially be resolved by a higher weighting of the L1-loss.

4.2. Efficiency Analysis

We analyze model complexity in terms of trainable parameters and approximate computational load for a forward pass (Floating Point Operations to compute for 1 second of audio) with a hybrid approach, as well as the *real-time factor (RTF)* on a single core of an *Intel(R) Xeon(R) Gold 5317 CPU @ 3.00GHz*. Results are shown in Table 2. While the CONV models are most efficient, they are not competitive. The hybrid **GCN-SSM** offers the best balance of quality and complexity. While the GCN baseline is computationally cheaper, the **GCN-SSM**’s perceptual gains come at a modest GFLOP increase. Adding SSM layers increases the load, but small state sizes keep it manageable. All models are real-time capable on modern CPUs. We include the RTF for completeness but note that it is highly dependent on the kernel optimizations within PyTorch and doesn’t guarantee real-time suitability for a live audio plugin. To demonstrate the true real-time potential of SSMs, we benchmarked a single layer in C++ with the hyperparameters used in **GCN-SSM**. We average a speed-up by a factor of **275**, compared to its PyTorch equivalent, utilizing the *Cpp-NN* library from [12].

5. CONCLUSION

This paper has demonstrated the significant perceptual benefits of hybrid architectures for spring reverb emulation, combining recurrent structures with feedforward layers. Our primary finding is that the interleaved **GCN-SSM** model achieves superior perceptual and objective performance, confirmed by its highest MUSHRA score. The systematic

Model	Parameters	GFLOP ↓	RTF ↓
CONV	11.6k	0.52	0.05
CONV-SSM	15.4k	0.85	0.21
GCN	113.6k	5.00	0.19
GCN-O	157.9k	6.95	0.24
GCN-SSM	125.7k	5.94	0.58
GCN-SSM-O	181.9k	7.99	0.56

Table 2. Model size and efficiency. Computational load was measured on a single core of an Intel Xeon Gold 5317 CPU. GFLOP for 1 second of audio at 44.1 kHz.

ablation of architectures provides strong evidence that this performance stems from a functional division of labor: the convolutional backbone models the reverb’s dynamic IR, while the SSMs refine complex phase relationships. Notably, due to the inclusion of abrupt signal cutoffs in our dataset, the GCN and GCN-SSM were able to learn the relationship between the input envelope and the reverb’s transient response, accurately reproducing the characteristic “splash” effect.

The interaction of SSMs, gating mechanisms, and dilated convolutions may prove even more impactful for other signal processing tasks, such as speech synthesis, where phase coherence is critical. We have shown that designing an appropriate loss function remains challenging, and that time-domain metrics might not always reflect perceptual quality, as seen in the results of **GCN-SSM-O**, which achieved the highest MUSHRA score. Nevertheless, the composite loss aligned with overall model quality.

Finally, our work shows that high perceptual quality can be achieved efficiently. We showed that all architectures are real-time capable on modern CPUs, making the hybrid models promising candidates for deployment in practical audio plugins.

6. ACKNOWLEDGMENTS

This work is supported by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology, and Development, and the Christian Doppler Research Association

7. REFERENCES

- [1] S. Bilbao and J. Parker, “A virtual model of spring reverberation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 799–808, 2010.
- [2] J. S. Abel, D. P. Berners, S. C., and J. O. Smith, “Spring reverb emulation using dispersive allpass filters in a waveguide structure,” *Journal of The Audio Engineering Society*, 2006.
- [3] J. McQuillan, M. Van Walstijn, J. D. Parker, and M. Ortiz, “Physical modelling of a spring reverb tank incorporating helix angle, damping, and magnetic bead coupling,” *Journal of the Audio Engineering Society*, vol. 73, no. 6, pp. 312–330, 2025.
- [4] U. Zölzer, Ed., *DAFX: Digital Audio Effects*, Wiley, second edition, 2011.
- [5] F. Papaleo, X. Lizarraga-Seijas, and F. Font, “Evaluating neural networks architectures for spring reverb modelling,” in *Proceedings of the 27th International Conference on Digital Audio Effects*, Guildford, UK, 2024, pp. 301–309.
- [6] C. J. Steinmetz and J. D. Reiss, “Efficient neural networks for real-time modeling of analog dynamic range compression,” in *Proceeding of the 152nd Audio Engineering Society Convention*, 2022.
- [7] M. A. Martínez Ramírez, E. Benetos, and Joshua D. Reiss, “Deep learning for black-box modeling of audio effects,” *Applied Sciences*, vol. 10, no. 2, 2020.
- [8] C. J. Steinmetz and J. D. Reiss, “Steerable discovery of neural audio effects,” in *NeurIPS Workshop on Machine Learning for Creativity and Design*, 2021.
- [9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, 1997.
- [10] J. T.H. Smith, A. Warrington, and S. Linderman, “Simplified state space layers for sequence modeling,” in *The Eleventh International Conference on Learning Representations*, 2023.
- [11] A. Gu, K. Goel, and C. Ré, “Efficiently modeling long sequences with structured state spaces,” in *The Tenth International Conference on Learning Representations*, 2022.
- [12] M. Bittner, D. Schnöll, M. Wess, and A. Jantsch, “Efficient and interpretable raw audio classification with diagonal state space models,” *Machine Learning*, vol. 114, 2025.
- [13] 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.
- [14] 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, pp. 30, 2025.
- [15] D. Dallinger, M. Bittner, D. Schnöll, M. Wess, and A. Jantsch, “Piano-SSM: Diagonal state space models for efficient midi-to-raw audio synthesis,” in *Proceedings of the 28th International Conference on Digital Audio Effects*, 2025, pp. 449–456.
- [16] K. Goel, A. Gu, C. Donahue, and C. Ré, “It’s raw! audio generation with state-space models,” in *International Conference on Machine Learning*, 2022.
- [17] Y. Zhang, G. Kolkman, and H. Watanabe, “Phase repair for time-domain convolutional neural networks in music super-resolution,” in *Proceedings of the 21st Sound and Music Computing Conference*, 2024, pp. 467–472.
- [18] M. A. Martínez Ramírez, E. Benetos, and J. D. Reiss, “Modeling plate and spring reverberation using a dsp-informed deep neural network,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, p. 241–245.
- [19] H. Pedroza, W. Abreu, R. M. Corey, and I. R. Roman, “Leveraging Electric Guitar Tones and Effects to Improve Robustness in Guitar Tablature Transcription Modeling,” in *Proceedings of the 27th International Conference on Digital Audio Effects*, 2024, pp. 143–146.
- [20] M. Comunità, C. J. Steinmetz, H. Phan, and J. D. Reiss, “Modelling black-box audio effects with time-varying feature modulation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.
- [21] C. J. Steinmetz and J. D. Reiss, “auraloss: Audio focused loss functions in PyTorch,” in *Digital Music Research Network One-day Workshop*, 2020.
- [22] M. Schoeffler, S. Bartoschek, F.R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, “webmushra — a comprehensive framework for web-based listening tests,” *Journal of Open Research Software*, vol. 6, 2018.