SOLID STATE BUS-COMP: A LARGE-SCALE AND DIVERSE DATASET FOR DYNAMIC RANGE COMPRESSOR VIRTUAL ANALOG MODELING

Yicheng Gu [12], Runsong Zhang*[2], Lauri Juvela[1] and Zhizheng Wu[2]*

[1]Acoustics Lab, Aalto University, Espoo, Finland
[2]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
yicheng.gu@aalto.fi

ABSTRACT

Virtual Analog (VA) modeling aims to simulate the behavior of hardware circuits via algorithms to replicate their tone digitally. Dynamic Range Compressor (DRC) is an audio processing module that controls the dynamics of a track by reducing and amplifying the volumes of loud and quiet sounds, which is essential in music production. In recent years, neural-network-based VA modeling has shown great potential in producing high-fidelity models. However, due to the lack of data quantity and diversity, their generalization ability in different parameter settings and input sounds is still limited. To tackle this problem, we present Solid State Bus-Comp, the first large-scale and diverse dataset for modeling the classical VCA compressor — SSL 500 G-Bus. Specifically, we manually collected 175 unmastered songs from the Cambridge Multitrack Library. We recorded the compressed audio in 220 parameter combinations, resulting in an extensive 2528-hour dataset with diverse genres, instruments, tempos, and keys. Moreover, to facilitate the use of our proposed dataset, we conducted benchmark experiments in various open-sourced black-box and grey-box models, as well as white-box plugins. We also conducted ablation studies in different data subsets to illustrate the effectiveness of the improved data diversity and quantity. The dataset and demos are on our project page: https://www.yichenggu.com/SolidStateBusComp/.

1. INTRODUCTION

Virtual Analog (VA) modeling aims to simulate analog audio devices digitally. Dynamic Range Compressor (DRC) is an audio processing module that compresses the dynamics of a track by reducing and amplifying the volumes of loud and quiet sounds, which is essential in music production [1]. VA modeling on DRC is important, but is always considered to be challenging due to its characteristics: non-linear and long temporal dependency.

To model an analog compressor, early DSP-based methods utilized white-box models. Such a model generally comprises a gain computer and a level detector with different algorithm designs [2, 1], which have been well-studied over the years. Apart from this, recent works have also been proposed to explore other potential improvements like increasing computational efficiency [3, 4] and integrating machine-learning techniques for automatic mixing [5, 6]. These developments have led to various achievements in modeling both the entire device [7] and specific components [8, 9].

Although these white-box techniques can deliver high-quality modeling over different devices, the involvement of human experts is often needed, making it hard to automate the modeling process. In recent years, neural-network-based black-box models have developed a lot due to their superior ability to model analog devices in a data-driven way. To be specific, [10] first proposed an autoencoder model to model various audio effects. [11] utilized the long short-term memory (LSTM) model for optimizing the long-term dependencies, followed by [12] to further expand into the hyper recurrent neural network (RNN) model with an in-depth comparison between RNN and LSTM models. To utilize the advantages of convolutional neural networks (CNNs), [13] first employed the WaveNet [14] structure on digital audio effects. Based on this work, [15] proposed a temporal convolutional network (TCN) with larger receptive fields and huge dilation factors, while [16] further improved this architecture by integrating the feature-wise linear modulation (FiLM) [17] layers in modeling the parameter conditions. State Space Model (SSM) [18] is another technique to model long-term dependent time series via decomposing a dynamic system into structured state variables. [19] first employed the S4 blocks in VA modeling, obtaining outstanding performances, followed by [20] further adopting the latest S6 model [21]. With the development of differentiable digital signal processing (DDSP) [22], works are also proposed to integrate the DSP models' explainability and efficiency with neural networks. For instance, [23] proposed differentiable biquad filters for deep learning applications, followed by [24] integrating them with Koopman Networks [25] to operate in a higher-dimensional state space. These advances have also made the neural grey-box models viable. In particular, [26] utilized the classic white-box DRC [1] design with multilayer perceptrons (MLPs) predicting the parameters in each time frame, followed by [27] to further simplify the model into parametric Gains for compression and supplementary EQs for non-linear distortion.

Despite the rapid development of VA models, the publicly available datasets are still scarce, with limited data quantity and diversity. Table 1 illustrates the details of the existing datasets regarding DRC. Specifically, early attempts [28] primarily consist of processed short instrument and test signal recordings in a specific parameter setting, tailored for trivial non-parameteric models. SignalTrain [10] first proposed a parametric dataset in modeling the optical compressor LA-2A. It used various randomly generated test signals and a few instrument recordings as the input signals and recorded 20 equally sampled parameter combinations. After that, [29] proposed the CL-1B dataset with real-world recordings as inputs with more parameter combinations. Recent works like [20] also presented datasets with more diverse devices but often with limited data scale and parameter combinations. Such limitations will significantly constrain the model's performance, especially when encountering real-world recordings and unseen parameters.

* Equal Contribution

Table 1: A comparison of Solid State Bus-Comp with existing VA modeling datasets regarding DRC.

Device	Duration (hour)	Type	Parameters	Range	Combinations
UA 6176 Limiter [28]	0.66	Transistor-Based Limiter	Attack Release Input Level Output Level Ratio	800 μs 1100 ms 4 7 All	1
Ampeg Opto Comp [39]	3.61	Optical Compressor	Compression Release Level	[3, 10] [1, 10] s 6	5
Flamma FC21 [39]	3.61	Optical Compressor	Comp EQ Volumn	[1, 10] [1, 10] 10	5
Yuer RF-10 [39]	3.61	OTA Compressor	Attack Sustain level	[1, 10] ms [1, 10] ms 10	6
Teletronix LA-2A [10]	48.63	Optical Compressor	Peak Reduction Switch Mode	[0, 100] [Compressor, Limiter]	20
TubeTech CL-1B [29]	37.54	Optical Compressor	Threshold Attack Release Ratio	[-40, 0] dB [5, 300] ms [0.005, 10] s 1:[1, 10]	108
SSL 500 G-Bus-Comp (ours)	2528.53	VCA Compressor	Threshold Attack Release Ratio	[-40, 0] dB [0.1, 30] ms [0.1, 1.6] s 1:[1.5, 10]	220

Data scaling has been shown to be effective in many audio-related areas [30, 31, 32, 33, 34]. For instance, Mert [30] utilized a music mixture of 160K hours to scale up a self-supervised representation learning model with 330M parameters, obtaining outstanding performance in music information retrieval; Yue [35] constructed a 650K hours music mixture to train a 7B parameter model for music generation, obtaining state-of-the-art (SOTA) performance; Stable Audio [31] collected 73k hours of audio recordings, leading to SOTA audio generation model with 1B parameters; Emilia [36, 37] presented a 101K hours open-sourced speech dataset, facilitating SOTA speech generations models [32, 38].

Following these previous works, this work presents Solid State Bus-Comp, the first large-scale and diverse dataset for modeling the SSL 500 G-Bus Compressor [1]. Specifically, we manually selected 175 unmastered real-world songs from the Cambridge Multitrack Library [2] and recorded the compressed signals in 220 parameter combinations, which results in an extensive 2528-hour dataset with diverse genres, instruments, tempos, and keys. To facilitate the use of our dataset, we conducted benchmarking experiments on various open-sourced black-box and grey-box models, as well as available white-box plugins. We also conducted ablation studies on data subsets with different amounts of songs and data scales to illustrate the effectiveness of the improved data diversity and quantity.

2. SOLID STATE BUS-COMP

This section provides the construction details, statistics, and analysis of our proposed Solid State Bus-Comp dataset.

2.1. Dataset Construction

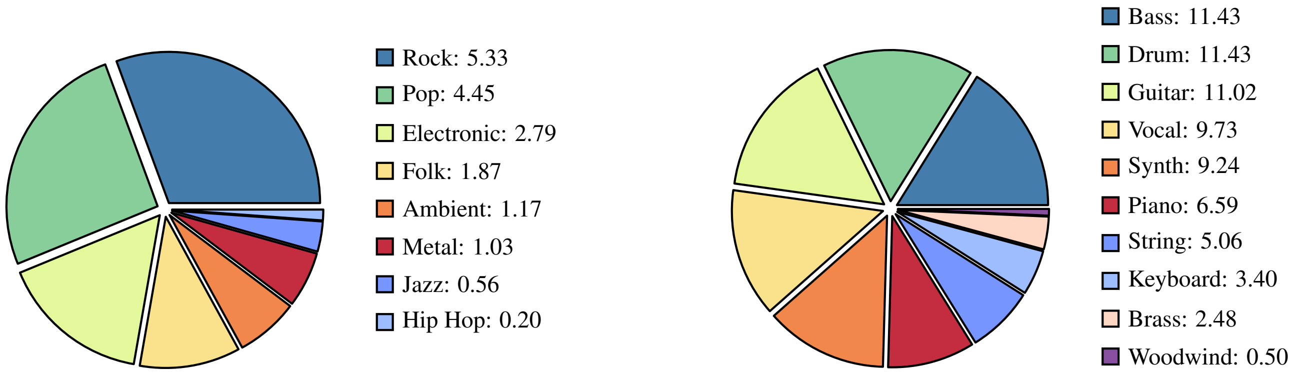
Solid State Bus-Comp comprises unmastered songs with different genres, instruments, tempos, and keys processed with varying compression parameters. In particular, we manually selected 175 unmastered songs from the Cambridge Multitrack Library [2]. We used Reaper [3] as the Digital Audio Workstation (DAW) to process the data automatically. Specifically, we used the RME Fireface UFX+ [4] as the external audio interface and connected it to the ReaInsert. Then, we wrote a ReaScript to automatically send and receive signals from the hardware compressor via the audio interface. To match the level between the DAW and hardware compressor, we normalized all songs to -12 dB and applied a 5 dB input boost and a 5 dB output attenuation. We manually selected 144 widely used parameter combinations for processing after consulting six professional mastering engineers, which are: threshold [-28, -24, -20, -16], attack [0.1, 0.3, 1, 3], release [0.1, 0.4, 0.8, auto], ratio [2, 4, 10]. We additionally recorded 76 other randomly selected combinations as supplementary edge cases. All the audio data was recorded at a sampling rate of 44.1 kHz.

[1] `https://solidstatelogic.com/products/stereo-bus-compressor-module`

[2] `https://www.cambridge-mt.com/ms/mtk/`

[3] `https://www.reaper.fm/`

[4] `https://rme-audio.de/fireface-ufx.html`



(a) Genre

(b) Instrument

Figure 1: Duration statistics (hours) of the unmastered songs used as input signals in Solid State Bus-Comp by genres and instruments.

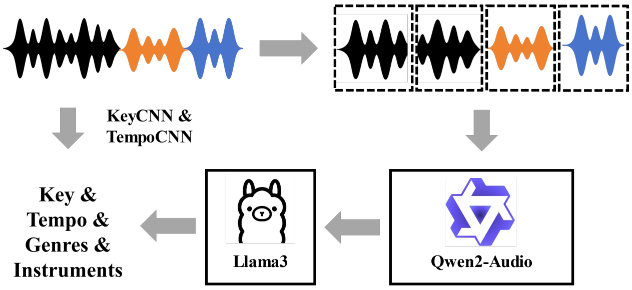


Figure 2: The annotation pipeline of Solid State Bus-Comp. We utilized various pre-trained models to obtain information on each song's key, tempo, genre, and instrument.

2.2. Dataset Statistics

We utilized various pre-trained models to annotate our data, as illustrated in Fig. 2. Specifically, we used the KeyCNN [5] and TempoCNN model [6] proposed in [40] [7] to obtain the global music tempo and key information. We split each song into a series of 10s segments and used the Qwen2-Audio [41] [8] to annotate each segment's content, which will then be fed to a Llama3 [42] [9] model to organize the genres and instruments of the whole song.

The statistical results of Solid State Bus-Comp on genres, instruments, tempos, and keys are illustrated in Fig. 1 and Fig. 3. From these results, we can conclude that 1) The majority of genres in our dataset are Rock, Pop, Electronic, and Folk, with a small amount of other uncommon ones like Ambient, Metal, Jazz and Hip Hop; 2) Most used instruments in our dataset are Bass, Drum, Guitar, Vocal, and Synth, with a considerable amount of Piano, String, Keyboard, and Brass. Niche instruments, like Woodwind, are also presented in the dataset; 3) Songs in our dataset are within the range of 70-160 beats per minute (BPM), and the majority of songs are distributed around 110-130 BPM; 4) Most songs in our dataset are in C, D, E, F, G, and A Majors, with a small number of remaining songs evenly distributed across other keys.

2.3. Dataset Analysis

Unlike existing datasets, which primarily utilize noises and analysis signals, Solid State Bus-Comp comprises a collection of diversified real-world unmastered songs as the input signals. To quantify this diversity, we use self-supervised learning (SSL) models to investigate and compare their differences in acoustic and semantic feature spaces, following [36], [37], and [33].

Specifically, to analyze the diversity of acoustic features, we leveraged a pre-trained MERT [30] [10] model to extract the acoustic representation (the 12th layer is used), which captures various acoustic characteristics such as timbre, style, key, etc. For the semantic diversity analysis, we employed a pre-trained w2v-BERT model [43] [11] to generate semantic representations (the last layer is used), capturing melody, lyrics, rhythm, etc. We then applied the Principal Component Analysis (PCA) algorithm to reduce the dimensionality of these representations to two. As illustrated in Fig. 4, most sample points in existing datasets are centered in two distant clusters, where the compact one represents the noise signals, and the diffused one represents the test signals (sine, square, triangle waves, and their combinations), and only a few points scattered aside, representing the real-world instrument recordings. Compared with the existing datasets, Solid State Bus-Comp exhibits a broader dispersion in the cluster representing real-world recordings, indicating richer acoustic and semantic characteristic coverage.

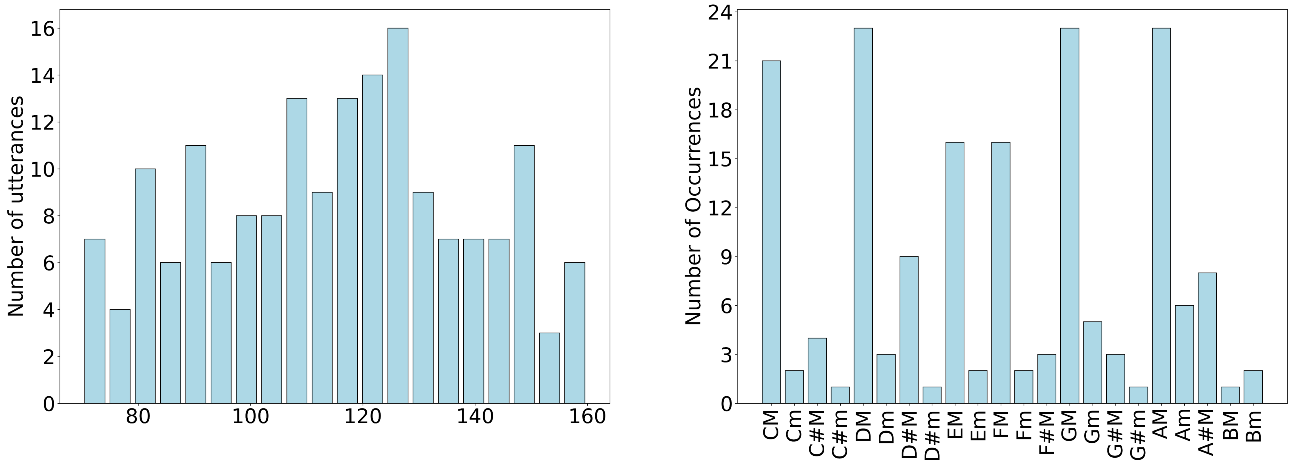
3. EXPERIMENTS

In this section, we conducted benchmark experiments to verify the effectiveness and facilitate the use of Solid State Bus-Comp. We also conducted ablation studies on different data subsets to illustrate the effectiveness of improved data scale and diversity.

3.1. Experiment Setup

Data Split and Processing: For the train and evaluation data split, we randomly selected 112 songs as the train set and used the remaining 63 songs as the test set. We used our manually selected 144 parameter combinations for training and the seen test distribution. The remaining 76 parameter combinations are used as the unseen test distribution to assess the generalization ability.

[5] https://github.com/hendriks73/key-cnn
[6] https://github.com/hendriks73/tempo-cnn
[7] https://github.com/hendriks73/directional_cnns
[8] https://huggingface.co/Qwen/Qwen2-Audio-7B
[9] https://huggingface.co/meta-llama
[10] https://huggingface.co/m-a-p/MERT-v1-330M
[11] https://huggingface.co/facebook/w2v-bert-2.0



(a) Tempo

(b) Key

Figure 3: Tempo and Key statistics (occurrences) of the unmastered songs used as input signals in our proposed Solid State Bus-Comp. Tempo is in beats per minute (BPM). "M" denotes for "Major" and "m" denotes for "Minor".

Training Schedules: All the models are trained using the AdamW [44] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a initial learning rate of 0.005. The ReduceLROnPlateau Scheduler is used with a factor of 0.5 and a patience of 10000 steps. All the experiments are conducted on a single NVIDIA H200 GPU with a batch size 16 and num workers of 16 for 500K steps. We use the Truncated Backpropagation Through Time (TBPTT) [45] with a 0.01s segment length (4410 samples) to reduce memory costs while maintaining long-term dependencies.

Baselines and Configurations: We use the NablAFx [27] toolbox for conducting benchmarking experiments on baseline systems. Specifically, we use LSTM [11], TCN [15], GCN [16], and S4 [19] for black-box models. The LSTM model is conditioned on direct concatenation (Concat) or time-varying concatenation (TVConcat) [39]. The TCN, GCN, and S4 models are conditioned on FiLM [17], temporal FiLM (TFiLM) [46], tiny temporal FiLM (TTFiLM) [39], and time-varying temporal FiLM (TVFiLM) [39]. We use GreyBoxDRC [26] and two compressor simulation chains proposed in ToneTwist [39] for grey-box models with the original configurations. For commercial plugins, we utilize the available models from Solid State Logic[12], Softube[13], Overloud[14], and PSPaudioware[15]. To facilitate reproducible research, all of the modified code and the pre-trained models can be accessed via [16].

Evaluation Metrics: We use the Amphion [47] toolkit for objective evaluation. We use the L1 and Multi-Resolution STFT losses to evaluate the time and frequency-domain errors following ToneTwist [39]. We additionally report the number of trainable parameters to show the model size.

3.2. Black-Box Methods

Table 2 illustrates the benchmarking results on black-box methods. Several key observations can be made: 1) Regarding the effectiveness of parameter scaling, LSTM and TCN models consistently benefit from increased model size. In contrast, GCN and S4 models only improve when conditioned on TTFiLM or TVFiLM layers. We speculate that the baseline FiLM layers used in these models are not expressive enough, leading to degraded performance as model capacity increases. On the other hand, TFiLM is powerful but introduces too many parameters, which may cause training instability in larger models. 2) Regarding parameter efficiency across different models, the LSTM model with TVConcat at 8.0K parameters achieves competitive results compared to larger models, and the S4 models with TVFiLM and TTFiLM reach near SOTA performance under 12K parameters. In contrast, models using TFiLM layers often require significantly more parameters to achieve comparable performance, making them unsuited for resource-constrained environments. 3) Regarding different conditioning layers, LSTM models with TVConcat perform significantly better than with simple concatenation. For TCN, GCN, and S4 models, TVFiLM surprisingly achieves the best performance, highlighting the effectiveness of time-varying modulation in modeling analog compressors. TFiLM generally ranks second, followed closely by TTFiLM, which offers a favorable trade-off between performance and parameter efficiency. 4) Regarding different model types, GCN consistently outperforms other architectures, demonstrating the strength of WaveNet-style dilated convolutions. S4 and TCN models with TTFiLM or TVFiLM also perform well. Notably, LSTM models with TVConcat outperform many other baselines, emphasizing the importance of temporal conditioning. 5) Regarding the generalization ability to unseen test scenarios, LSTM models with TVConcat and TCN, GCN, and S4 models with TFiLM, TTFiLM, and TVFiLM maintain strong performance on both seen and unseen parameter settings. In contrast, models using simpler conditioning layers exhibit noticeable performance drops under unseen testing senarios.

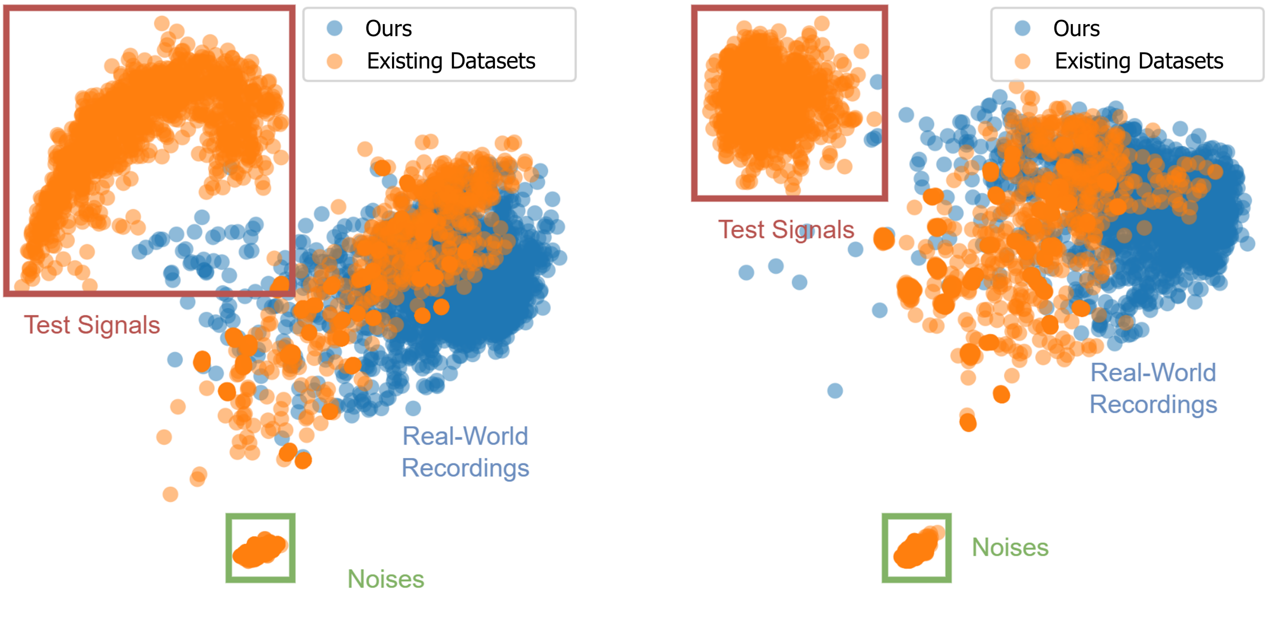
[12] https://store.solidstatelogic.com/plug-ins/ssl-native-bus-compressor-2

[13] https://www.softube.com/bus-processor

[14] https://www.overloud.com/products/comp-g

[15] https://www.pspaudioware.com/products/psp-busspressor

[16] https://drive.google.com/drive/folders/1zf5hnF7XGRW-poo_cqjQthKBeAZx33gd



(a) Acoustic Diversity

(b) Semantic Diversity

Figure 4: Comparison of acoustic and semantic diversities in input signals between Solid State Bus-Comp and the existing datasets. The plottings are obtained by applying the PCA algorithm to the SSL representations. We used MERT to extract acoustic embeddings and w2v-BERT 2.0 to extract semantic embeddings. For existing datasets, the compact cluster represents random noises, the diffused cluster represents test signals (sine, square, triangle waves, etc), and the remaining scattered points represent real-world recordings.

3.3. Grey-Box Methods

The benchmarking results on grey-box models are presented in Table 3. It can be observed that: 1) Regarding different gain computer models, static gain with a soft knee generally performs better with different level detectors. This aligns with the analog design of the SSL G-Bus compressor, which employs a soft knee where the knee width is automatically computed based on the threshold and ratio [17]. 2) For different level detector implementations, the switching one-pole filter achieves the best overall performance, followed by the standard one-pole filter. In contrast, the RNN-modulated one-pole filter performs worse. We speculate that this is due to the relatively simple design of the VCA compressor's level detection circuit, which is different from the LA-2A that has strong non-linear distortion due to its optical components. Under this scenario, overly complex models like RNN-based detectors may overfit and lead to quality degradation. 3) egarding different test sets, a noticeable performance gap is observed between seen and unseen parameter settings. This can also be attributed to the changing compressor curve in the analog module [17], making it hard for grey-box models without explicit feedback mechanisms to capture that information. 4) In analog effect chain simulation, it is notable that the best performance is achieved using two parametric gain modules for compression and two parametric EQs for non-linear coloration. This illustrates the powerful learning ability of neural networks in loose conditions. Experiments also show adding a simple phase inversion module would damage the model performance since there are no phasers in the actual analog module, confirming its effectiveness and explainability.

3.4. White-Box Plugins

To evaluate the development of NN-based models and further illustrate the effectiveness of our proposed dataset, benchmarking results on white-box plugins are also reported, as shown in Table 4. Compared to these industry-standard plugins, a significant performance gap remains, particularly under extreme compression scenarios. This highlights that even the SOTA academic NN-based models still lag behind their commercial counterparts, which also illustrates the importance of our work since both model structure and datasets need to be improved for better performance.

3.5. Ablation Study

We also conducted ablation studies to illustrate the effectiveness of improving data quantity and diversity. We selected the GCN model conditioned with the TVFiLM layer as the baseline model and compared its performance when trained on different subsets. In particular, to control the data quantity, we fixed the number of total songs to 100 and control the length used to clip each song, resulting in 5 subsets from 3 minutes to 500 hours; to investigate the data diversity, we fixed the total data quantity to 50 hours and control the number of total songs with the adjusted clip lengths, resulting in 5 subsets from 5 songs to 100 songs. The detailed results are illustrated in Table 5. It can be observed that 1) increasing the data quantity steadily improves the model performance from 3 minutes to 500 hours, with the 50 hours as the division line for significant improvement, which is also confirmed by previous works [48]. 2) Increasing the data diversity is effective when there are only a few songs, and the improvement will be saturated until there are 50 different songs, especially in the unseen parameter settings.

[17] https://www.solidstatelogic.com/assets

Table 2: Benchmarking results of existing parametric black-box methods. The best and second best results are **bold** and underlined.

System	Configuration	Condition	#Params	L1 (↓)		M-STFT (↓)	
				Seen	Unseen	Seen	Unseen
LSTM [11]	32 Channels	Concat	5.0K	0.0290	0.0239	0.3954	0.4644
		TVConcat	8.0K	0.0030	0.0028	0.3631	0.4523
	96 Channels	Concat	39.7K	0.0274	0.0237	0.4732	0.8123
		TVConcat	45.7K	**0.0028**	0.0029	0.4256	0.5483
TCN [15]	5 Blocks 7 Kernel 4 Dilation	FiLM	15.0K	0.0296	0.0251	0.5432	0.8647
		TFiLM	42.0K	0.0066	0.0056	0.3755	0.4492
		TTFiLM	17.3K	0.0271	0.0224	0.3903	0.4953
		TVFiLM	17.7K	0.0252	0.0224	0.5957	0.9704
	10 Blocks 3 Kernel 2 Dilation	FiLM	20.1K	0.0088	0.0079	0.5158	0.6959
		TFiLM	76.4K	0.0080	0.0067	0.3731	0.4427
		TTFiLM	27.0K	0.0260	0.0215	0.3804	0.5057
		TVFiLM	22.8K	0.0083	0.0069	0.3819	**0.3983**
GCN [16]	5 Blocks 7 Kernel 4 Dilation	FiLM	29.0K	0.0271	0.0223	0.4760	0.5527
		TFiLM	146.0K	0.0041	0.0034	0.3713	0.4045
		TTFiLM	31.6K	0.0066	**0.0024**	0.3817	0.5766
		TVFiLM	31.7K	0.0270	0.0226	0.3406	0.4147
	10 Blocks 3 Kernel 2 Dilation	FiLM	40.5K	0.0241	0.0200	0.6757	0.6346
		TFiLM	278.0K	0.0267	0.0220	0.3497	0.4438
		TTFiLM	48.0K	0.0063	**0.0024**	0.3549	0.5766
		TVFiLM	43.2K	0.0272	0.0226	**0.3238**	0.4456
S4 [19]	4 Blocks 4 State Dimension	FiLM	8.9K	0.0287	0.0246	0.8044	1.0532
		TFiLM	30.0K	0.0277	0.0230	0.3576	0.4973
		TTFiLM	10.2K	0.0030	0.0030	0.3884	0.4689
		TVFiLM	11.6K	0.0283	0.0237	0.3898	0.5842
	8 Blocks 32 State Dimension	FiLM	29.7K	0.0103	0.0102	1.0552	1.2474
		TFiLM	74.3K	0.0046	0.0043	0.4961	0.6098
		TTFiLM	34.8K	0.0265	0.0225	0.4665	0.5898
		TVFiLM	32.4K	0.0030	0.0031	0.3480	0.4930

4. CONCLUSION

In conclusion, this paper presents Solid State Bus-Comp, the first extensive and diverse dataset for DRC VA modeling. Our dataset comprises 2528 hours of processed unmastered songs in 220 parameter combinations with diverse genres, instruments, tempos, and keys. We provide benchmarking results on various open-sourced black-box and grey-box models, as well as available white-box plugins to facilitate the use of our dataset. We also provide ablation experiment results on different data subsets to illustrate the effectiveness of the improved data scale and quantity.

5. ACKNOWLEDGMENT

We acknowledge the computational resources provided by the Aalto Science-IT project. We acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LUMI, hosted by CSC (Finland) and the LUMI consortium through a EuroHPC Regular Access call. This work is also supported by the 2023 Shenzhen stability Science Program, the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2023ZT10X044), and the Shenzhen Science and Technology Program (ZDSYS20230626091302006)

6. REFERENCES

[1] Dimitrios Giannoulis, Michael Massberg, and Joshua D Reiss, "Digital dynamic range compressor design—A tutorial and analysis," *Journal of the Audio Engineering Society*, vol. 60, no. 6, pp. 399–408, 2012.

[2] Guy W McNally, "Dynamic range control of digital audio signals," *Journal of the Audio Engineering Society*, vol. 32, no. 5, pp. 316–327, 1984.

[3] Germán Ramos, "Block processing strategies for computationally efficient dynamic range controllers," in *Proc. Int. Conf. Digital Audio Effects*, 2011, pp. 253–256.

[4] Leo McCormack and Vesa Välimäki, "FFT-based dynamic range compression," in *Sound Music Comput. Conf.*, 2017, pp. 42–49.

[5] Dimitrios Giannoulis, Michael Massberg, and Joshua D Reiss, "Parameter automation in a dynamic range compressor," *Journal of the Audio Engineering Society*, vol. 61, no. 10, pp. 716–726, 2013.

[6] Jacob A Maddams, Saoirse Finn, and Joshua D Reiss, "An autonomous method for multi-track dynamic range compres-

Table 3: Benchmarking results of existing grey-box methods. The best and second best results are **bold** and underlined.

System	**Signal Chain**			**#Params**	**L1 (↓)**		**M-STFT (↓)**	
	Static Gain	**Make-Up Gain**	**Level Detector**		**Seen**	**Unseen**	**Seen**	**Unseen**
GreyBoxDRC [26]	Soft Knee	Static Gain	One-Pole	0.6K	0.0076	0.0066	1.0046	1.1312
			Switching One-Pole	0.6K	0.0067	0.0072	0.8108	1.2388
			RNN Mod. One-Pole	0.7K	0.0076	0.0070	1.0066	1.2251
	Hard Knee	GRU	One-Pole	0.8K	0.0062	0.0074	1.1072	1.5134
			Switching One-Pole	0.8K	0.0059	0.0061	0.8758	1.0888
			RNN Mod. One-Pole	0.9K	0.0061	0.0070	1.1218	1.6492
ToneTwist [39]	PEQ → Gain → PEQ → Gain			1.6K	**0.0034**	**0.0034**	**0.4098**	**0.6004**
	PEQ → Phase Inversion → Gain → PEQ → Gain			2.0K	0.0200	0.0168	1.5964	1.4596

Table 4: Benchmarking results of existing commercial plugins. The best and second best results are **bold** and underlined.

System	**L1 (↓)**		**M-STFT (↓)**	
	Seen	**Unseen**	**Seen**	**Unseen**
Solid State Logic	0.0322	0.0175	0.4489	0.2943
Softube	0.0448	0.0237	0.7069	0.4546
Overloud	0.0326	0.0176	0.4738	0.3253
PSPaudioware	**0.0269**	**0.0145**	**0.3047**	**0.2184**

Table 5: Ablation results of the GCN model trained on different data subsets. The best and second best results of every column in each setting are **bold** and underlined.

#Songs	**Duration (hour)**	**L1 (↓)**		**M-STFT (↓)**	
		Seen	**Unseen**	**Seen**	**Unseen**
100	0.05	0.0279	0.0262	0.4718	0.5699
	0.5	0.0278	0.0226	0.3649	0.4838
	5	0.0298	**0.0222**	0.3641	0.4539
	50	0.0273	0.0227	0.3245	0.4520
	500	**0.0272**	0.0226	**0.3238**	**0.4456**
5	50	0.0276	0.0231	0.4793	0.5876
10		0.0277	0.0230	0.4294	0.5159
25		0.0277	0.0227	0.3333	0.5030
50		0.0274	**0.0224**	0.3247	**0.4502**
100		**0.0273**	0.0227	**0.3245**	0.4520

sion," in *Proc. Int. Conf. Digital Audio Effects*, 2012, pp. 1–8.

[7] Oliver Kröning, Kristjan Dempwolf, and Udo Zölzer, "Analysis and simulation of an analog guitar compressor," *Proc. Int. Conf. Digital Audio Effects*, pp. 205–208, 2011.

[8] Felix Eichas and Udo Zölzer, "Modeling of an optocoupler-based audio dynamic range control circuit," in *Novel Optical Systems Design and Optimization XIX*, 2016, vol. 9948, pp. 47–62.

[9] Alessandro Ilic Mezza, Riccardo Giampiccolo, and Alberto Bernardini, "Data-Driven Parameter Estimation of Lumped-Element Models via Automatic Differentiation," *IEEE Access*, vol. 11, pp. 143601–143615, 2023.

[10] Scott H Hawley, Benjamin Colburn, and Stylianos I Mimilakis, "SignalTrain: Profiling audio compressors with deep neural networks," *arXiv:1905.11928*, 2019.

[11] Alec Wright, Eero-Pekka Damskägg, and Vesa Välimäki, "Real-time black-box modelling with recurrent neural networks," in *Proc. Int. Conf. Digital Audio Effects*, 2019.

[12] Yen-Tung Yeh, Wen-Yi Hsiao, and Yi-Hsuan Yang, "Hyper recurrent neural network: Condition mechanisms for black-box audio effect modeling," *arXiv:2408.04829*, 2024.

[13] Alec Wright, Eero-Pekka Damskägg, Lauri Juvela, and Vesa Välimäki, "Real-time guitar amplifier emulation with deep learning," *Applied Sciences*, vol. 10, no. 3, pp. 766, 2020.

[14] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al., "Wavenet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[15] Christian J Steinmetz and Joshua D Reiss, "Efficient neural networks for real-time modeling of analog dynamic range compression," *arXiv:2102.06200*, 2021.

[16] Marco Comunità, Christian J Steinmetz, Huy Phan, and Joshua D Reiss, "Modelling black-box audio effects with time-varying feature modulation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2023, pp. 1–5.

[17] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville, "FiLM: Visual Reasoning with a General Conditioning Layer," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32.

[18] Albert Gu, Karan Goel, and Christopher Ré, "Efficiently Modeling Long Sequences with Structured State Spaces," in *Proc. Int. Conf. Learn. Representations*, 2022.

[19] Hanzhi Yin, Gang Cheng, Christian J Steinmetz, Ruibin Yuan, Richard M Stern, and Roger B Dannenberg, "Modeling analog dynamic range compressors using deep learning and state-space models," *arXiv:2403.16331*, 2024.

[20] Riccardo Simionato and Stefano Fasciani, "Comparative study of recurrent neural networks for virtual analog audio effects modeling," *arXiv:2405.04124*, 2024.

[21] Albert Gu and Tri Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv:2312.00752*, 2023.

[22] Jesse H. Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts, "DDSP: Differentiable Digital Signal Processing," in *Proc. Int. Conf. Learn. Representations*, 2020.

[23] Boris Kuznetsov, Julian D Parker, and Fabián Esqueda, "Differentiable IIR filters for machine learning applications," in *Proc. Int. Conf. Digital Audio Effects*, 2020, pp. 297–303.

[24] Ville Huhtala, Lauri Juvela, and Sebastian J. Schlecht, "KLANN: Linearising Long-Term Dynamics in Nonlinear Audio Effects Using Koopman Networks," *IEEE Signal Process. Lett.*, vol. 31, pp. 1169–1173, 2024.

[25] Bethany Lusch, J Nathan Kutz, and Steven L Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, no. 1, pp. 4950, 2018.

[26] Alec Wright and Vesa Valimaki, "Grey-box modelling of dynamic range compression," in *Proc. Int. Conf. Digital Audio Effects*, 2022, pp. 304–311.

[27] Marco Comunità, Christian J Steinmetz, and Joshua D Reiss, "NablAFx: A Framework for Differentiable Black-box and Gray-box Modeling of Audio Effects," *arXiv:2502.11668*, 2025.

[28] Marco A Martínez Ramírez, Emmanouil Benetos, and Joshua D Reiss, "Deep learning for black-box modeling of audio effects," *Applied Sciences*, vol. 10, no. 2, pp. 638, 2020.

[29] Riccardo Simionato and Stefano Fasciani, "Fully conditioned and low-latency black-box modeling of analog compression," in *Proc. Int. Conf. Digital Audio Effects*, 2023.

[30] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger B. Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu, "MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training," in *Proc. Int. Conf. Learn. Representations*, 2024.

[31] Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons, "Stable audio open," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2025, pp. 1–5.

[32] Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu, "MaskGCT: Zero-shot text-to-speech with masked generative codec transformer," in *Proc. Int. Conf. Learn. Representations*, 2024.

[33] Yicheng Gu, Chaoren Wang, Junan Zhang, Xueyao Zhang, Zihao Fang, Haorui He, and Zhizheng Wu, "SingNet: Towards a Large-Scale, Diverse, and In-the-Wild Singing Voice Dataset," *OpenReview*, 2024.

[34] Yicheng Gu, Chaoren Wang, Zhizheng Wu, and Lauri Juvela, "Neurodyne: Neural Pitch Manipulation with Representation Learning and Cycle-Consistency GAN," 2025.

[35] Ruibin Yuan, Hanfeng Lin, Shuyue Guo, Ge Zhang, Jiahao Pan, Yongyi Zang, Haohe Liu, Yiming Liang, Wenye Ma, Xingjian Du, et al., "YuE: Scaling Open Foundation Models for Long-Form Music Generation," *arXiv:2503.08638*, 2025.

[36] Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu, "Emilia: An Extensive, Multilingual, and Diverse Speech Dataset for Large-Scale Speech Generation," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2024, pp. 885–890.

[37] Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al., "Emilia: A Large-Scale, Extensive, Multilingual, and Diverse Dataset for Speech Generation," *arXiv:2501.15907*, 2025.

[38] Xueyao Zhang, Xiaohui Zhang, Kainan Peng, Zhenyu Tang, Vimal Manohar, Yingru Liu, Jeff Hwang, Dangna Li, Yuhao Wang, Julian Chan, et al., "Vevo: Controllable zero-shot voice imitation with self-supervised disentanglement," in *Proc. Int. Conf. Learn. Representations*, 2025.

[39] Marco Comunità, Christian J Steinmetz, and Joshua D Reiss, "Differentiable Black-box and Gray-box Modeling of Nonlinear Audio Effects," *arXiv:2502.14405*, 2025.

[40] Hendrik Schreiber and Meinard Müller, "Musical tempo and key estimation using convolutional neural networks with directional filters," *arXiv:1903.10839*, 2019.

[41] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al., "Qwen2-audio technical report," *arXiv:2407.10759*, 2024.

[42] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al., "The llama 3 herd of models," *arXiv:2407.21783*, 2024.

[43] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu, "w2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 244–250.

[44] Ilya Loshchilov and Frank Hutter, "Decoupled Weight Decay Regularization," in *Proc. Int. Conf. Learn. Representations*, 2019.

[45] Christopher Aicher, Nicholas J. Foti, and Emily B. Fox, "Adaptively Truncating Backpropagation Through Time to Control Gradient Bias," in *Conf. Uncertain. Artif. Intell.*, 2019, pp. 799–808.

[46] Marco Comunità, Christian J. Steinmetz, Huy Phan, and Joshua D. Reiss, "Modelling Black-Box Audio Effects with Time-Varying Feature Modulation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2023, pp. 1–5.

[47] Xueyao Zhang, Liumeng Xue, Yicheng Gu, Yuancheng Wang, Jiaqi Li, Haorui He, Chaoren Wang, Ting Song, Xi Chen, Zihao Fang, Haopeng Chen, Junan Zhang, Tze Ying Tang, Lexiao Zou, Mingxuan Wang, Jun Han, Kai Chen, Haizhou Li, and Zhizheng Wu, "Amphion: An Open-Source Audio, Music and Speech Generation Toolkit," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, 2024, pp. 879–884.

[48] Lauri Juvela, Eero-Pekka Damskägg, Aleksi Peussa, Jaakko Mäkinen, Thomas Sherson, Stylianos I Mimilakis, Kimmo Rauhanen, and Athanasios Gotsopoulos, "End-to-end amp modeling: from data to controllable guitar amplifier models," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2023, pp. 1–5.