TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Andrei Guzovski 232215IAPM

DEVELOPMENT OF A TOOLKIT FOR AGGREGATION AND GENERATION OF REAL-WORLD GUITAR AUDIO DATA

Master's Thesis

Supervisor: Uljana Reinsalu
PhD

Tallinn 2025

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Andrei Guzovski 232215IAPM

TÖÖRIISTAKOMPLEKTI ARENDAMINE REAALMAAILMA KITARRIHELI ANDMETE KOONDAMISEKS JA GENEREERIMISEKS

Magistritöö

Juhendaja: Uljana Reinsalu
PhD

Tallinn 2025

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Andrei Guzovski

12.05.2025

Abstract

Research on removing effects from guitar recordings is hampered by the lack of large-scale, diverse processed audio resources. Without comprehensive datasets, deep learning models struggle to generalize beyond controlled laboratory recordings, limiting their usefulness in tasks such as restoring archival performances, isolating instrument tracks for remixing, and improving automatic transcription and music analysis.

This thesis presents a toolkit that automates the aggregation and preprocessing of real-world guitar audio. It unifies multiple public collections, implements efficient workflows for audio concatenation, metadata-based segmentation, and configurable silence removal, and offers optional pre-rendering for computationally intensive experiments. Built on modern frameworks, the toolkit integrates directly with standard deep learning training pipelines to ensure reproducibility, scalability, and ease of use.

To validate the toolkit, a series of controlled experiments examined the impact of different normalization methods, the blending of acoustic and electric recordings, the inclusion of monophonic and polyphonic textures, the benefits of combining multiple source collections, and two augmentation strategies—simple waveshaping and neural network–based simulation. Model performance was evaluated in terms of improvements in signal clarity and reduction of distortion artifacts.

Results reveal that perceptual loudness normalization, when applied consistently during training and evaluation, yields the greatest gains in output clarity. Models trained on aggregated, varied recordings consistently outperformed those using single-source data. Moreover, advanced neural network–based augmentation delivered larger relative improvements than baseline distortion, indicating that realistic effect simulation is crucial for generalization. This toolkit thus provides a robust, extensible platform and practical guidelines that advance guitar audio effect removal and support applications in audio restoration, creative remixing, and music information retrieval.

The thesis is written in English and is 75 pages long, including 7 chapters, 13 figures and 15 tables.

Annotatsioon

Tööriistakomplekti arendamine reaalmaailma kitarriheli andmete koondamiseks ja genereerimiseks

Kitarrisalvestustelt efektide eemaldamise uurimist takistab suures mahus mitmekülgsete heliallikate puudus. Ilma mitmekülgsete andmekogudeta on süvaõppemudelitel keeruline üldistada väljaspool kontrollitud laboritingimusi, mis piirab nende rakendatavust arhiivsalvestiste taastamisel, pilliradade eraldamisel remiksimiseks ning automaatse transkriptsiooni ja muusikaanalüüsi täiustamisel.

Käesolev magistritöö esitleb tööriistakomplekti, mis automatiseerib reaalse maailma kitarriaudio kogumise ja eeltöötluse. See liidab mitu avalikku andmekogu, rakendab tõhusaid töövooge heli ühendamiseks, metainformatsiooni põhist segmenteerimiseks ja konfigureeritava vaikuse eemaldamiseks ning pakub valikulist eelrenderdamist arvutusmahukate katsete jaoks. Modernsetest raamistikest lähtudes integreerub tööriistakomplekt otse süvaõppe treeningahelatesse, tagades korratavuse, laiendatavuse ja kasutusmugavuse.

Tööriistakomplekti valideerimiseks viidi läbi rida kontrollitud eksperimente, milles uuriti erinevate normaliseerimismeetodite mõju, akustiliste ja elektriliste salvestuste segamist, monofoniliste ja polüfoniliste tekstuuride kaasamist, mitme allikakogu kombineerimise eeliseid ning kahte andmete suurendamise strateegiat – lihtsat lainekujude kujundamist ja neuronvõrkudel põhinevat simulatsiooni. Mudeleid hinnati signaali selguse paranemise ja moonutuste artefaktide vähenemise alusel.

Tulemused näitavad, et tajuline helitugevuse normaliseerimine annab treeningu ja hindamise käigus järjekindlalt suurimad selguse parandused, kui see on ühtlaselt rakendatud. Agregeeritud ja mitmekesistel salvestustel treenitud mudelid ületasid pidevalt üksikallikast pärinevate andmetega treenitud mudeleid. Lisaks näitas neuronvõrkudel põhinev andmete suurendamine suhtelisi suuremaid täiustusi võrreldes baasdistortsiooniga, mis viitab sellele, et realistlik efektisimulatsioon on generaliseerumise seisukohast kriitiline. See tööriistakomplekt pakub seega tugevat ja laiendatavat platvormi ning praktilisi juhiseid, mis edendavad kitarriefektide eemaldamist, toetavad audio restaureerimist, loomingulist remiksimist ja muusikainformatsiooni päringuid.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 75 leheküljel, 7 peatükki, 13 joonist, 15 tabelit.

List of Abbreviations and Terms

AMT	Automatic Music Transcription
DAFX	Digital Audio Effects
DAW	Digital Audio Workstation
DDD	Demucs-Discriminator-Declipper
dB	Decibels
dBFS	Decibels Full Scale
DI	Direct Input
DNN	Deep Neural Network
DSP	Digital Signal Processing
GAN	Generative Adversarial Network
GTT	Guitar Tablature Transcription
HPC	High-Performance Computing
LUFS	Loudness Units Full Scale
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MRSTFT	Multi-Resolution Short-Time Fourier Transform
NN	Neural Network
SI-SDR	Scale-Invariant Signal-to-Distortion Ratio
TCN	Time-Conditioned Network
Tanh	Tanh function based distortion (soft-clipping)
VST	Virtual Studio Technology

Table of Contents

List of Figures

List of Tables

1. Introduction

Musicians and audio engineers interact with sound in multiple ways daily, from transcribing intricate performances and learning new pieces, to creatively shaping timbres with audio effects, and composing using a variety of digital tools. In recent years, the rapid evolution of artificial intelligence (AI), particularly deep learning, has begun to revolutionize the field of audio signal processing, offering powerful new methods to assist with these tasks and unlock new creative possibilities. AI-driven tools are increasingly integrated into Digital Audio Workstations (DAWs), plugins, and standalone applications, aiding in tasks ranging from automated transcription and source separation to intelligent mixing and mastering.

Among the various manipulations of audio, the application of effects—such as distortion, delay, and reverb—is fundamental to modern music production, especially for instruments like the electric guitar where effects are integral to the characteristic sound. While these effects are powerful creative tools, the inverse problem—removing or "undoing" these effects to recover the original, clean signal—presents a significant technical challenge. The ability to effectively remove audio effects is highly valuable for numerous applications. For instance, it can enable the restoration of old or poorly processed recordings, facilitate remixing and remastering by isolating clean instrumental tracks, provide cleaner signals for music information retrieval (MIR) tasks like transcription or chord recognition, and serve as a pedagogical tool for understanding signal processing chains.

Despite its importance, developing robust deep learning models for audio effect removal, particularly for complex, non-linear effects like guitar distortion, faces two critical open problems. Firstly, there is a significant scarcity of large-scale, diverse, and consistently processed training data. While some datasets exist, they often lack the sheer volume, timbral variety, or specific clean/effected pairings needed to train models that generalize well to real-world audio. Secondly, evaluating the performance of these models on "real-life" audio with a wide array of effect types, parameter settings, and recording conditions remains a complex endeavor. This makes it difficult to benchmark progress and understand the true capabilities of developed systems. The practical uptake of advanced effect removal in industry tools, while growing, is often hampered by these data and evaluation bottlenecks.

This thesis directly addresses these challenges within the domain of guitar audio. Recognizing the critical need for better data resources and standardized processing, the overarching

aim of this work is to develop a comprehensive, engineeringly sound toolkit. This toolkit is designed to facilitate the aggregation of diverse existing guitar datasets and the generation of high-quality, consistently processed guitar audio data, specifically tailored for training and evaluating robust effect removal models and supporting related MIR tasks. By integrating heterogeneous datasets, implementing advanced and configurable normalization and augmentation techniques, and enabling rigorous cross-dataset evaluation methodologies, this thesis seeks to provide a replicable framework that can advance both academic research and practical applications in guitar audio processing.

1.1 Background

Guitar audio research has evolved significantly over the past decade, driven by both the creation of specialized datasets and the development of novel processing techniques. Early efforts, such as the *GuitarSet* dataset [1], laid the groundwork for automatic transcription by providing well-annotated acoustic recordings. Later collections like EGDB [2] and IDMT-SMT-Guitar [3] expanded the focus to include electric guitar recordings with diverse timbral properties. More recently, synthetic datasets such as SynthTab [4] and the comprehensive Guitar-TECHS [5] have been introduced to overcome data scarcity, enabling pre-training and fine-tuning strategies that improve cross-domain generalization. Fundamental to leveraging these diverse datasets is the issue of audio normalization. Traditional approaches like per-sample peak normalization lack perceptual consistency, while internal methods like Batch Normalization [6], common in deep learning, operate within the model rather than as a preprocessing step. Perceptual loudness normalization tools, such as `pyloudnorm` [7] implementing the ITU-R BS.1770 standard, are emerging as viable alternatives for consistent preprocessing, though their comparative benefits specifically for training deep audio models remain underexplored. Alongside normalization, data augmentation strategies are crucial for simulating real-world variability. These range from simple signal transformations (e.g., soft-clipping using libraries like `Pedalboard` [8]) and processing with commercial VST plugins [9] (offering realism but facing accessibility issues) to advanced neural network-driven methods [10], including community-driven models from projects like Guitar ML [11] and Neural Amp Modeler [12]. These developments set the stage for a comprehensive approach that not only aggregates heterogeneous guitar data but also systematically addresses the challenges posed by complex audio effects, particularly non-linear distortion.

1.2 Problem Statement

The primary objective of this thesis is to improve the generalization and robustness of deep learning models for guitar distortion removal by addressing three key challenges: data aggregation, normalization, and augmentation. Specifically, the research seeks to answer the following questions:

- What are the most effective methods to increase size and diversity of training data for guitar effect removal?
- Which normalization strategies best preserve characteristics of audio and ensure stable training for neural net models?
- How can both baseline synthetic and advanced neural network-based data augmentation techniques be leveraged to enhance model generalization for effect removal?
- What role does the categorization of guitar audio data play in the performance of distortion removal systems?

By answering these questions, this thesis aims to establish best practices for data processing that can enhance the performance of guitar distortion removal models across diverse, real-world scenarios.

The research object of this study is a comprehensive toolkit for the aggregation, normalization, categorization, and augmentation of guitar audio data, with the ultimate goal of recovering clean (dry) guitar signals from recordings affected by complex, non-linear distortion. Building upon established concepts in audio processing and deep learning, this work will employ rigorous cross-dataset validation to verify that the proposed methodologies yield reproducible and robust results. The research plan involves curating diverse datasets (specifically *GuitarSet* [1], EGDB [2], IDMT-SMT-Guitar [3], and Guitar-TECHS [5]), experimenting with various normalization techniques (peak vs. loudness normalization via `pyloudnorm` [7]), and developing augmentation pipelines that combine both baseline (*e.g.*, tanh distortion) and advanced neural approaches (using OpenAmp models [10]). State-of-the-art distortion removal models—specifically the RemFx framework [13] with a Hybrid Demucs architecture, comparable to methods like Lee *et al.* [9]—will be trained and evaluated using objective metrics (SI-SDR [14], MRSTFT [15]) as well as qualitative listening tests. Preliminary experiments have suggested that synthetic data augmentation can significantly reduce overfitting and improve transcription performance [4, 16], and this thesis extends these investigations to the domain of distortion removal. The proposed solution is both novel and highly relevant, as it addresses the scarcity of large-scale, diverse, and consistently processed guitar data for effect removal research, alongside the need for replicable, robust methodologies. Although the task is complex—requiring integration of

data engineering, signal processing, and advanced deep learning—it is well-suited for a master's thesis given its potential to make a substantial contribution to both research and industry.

1.3 Related Work

A substantial body of literature underpins the research in guitar audio processing and effect removal. Lee *et al.* [9] present a two-stage GAN-based approach for distortion removal combining spectral cleaning with neural vocoding, achieving strong performance but relying on large, proprietary datasets, limiting broader applicability. In contrast, general-purpose audio effect removal frameworks like RemFx [13] and audio effect chain estimation methods [17] aim for more comprehensive modeling of real-world processing pipelines. Additional contributions include blind extraction techniques for guitar effects [18] and approaches for real-time amplifier emulation [19, 20], blurring the lines between effect removal and application. Synthetic data methodologies, demonstrated in SynthTab [4] and the Open-Amp framework [10], show that incorporating large-scale simulated data enhances model robustness. Recent advances in zero-shot domain adaptation for transcription [21] also offer strategies to mitigate data scarcity. The repository compiled in AFX-Research [22] further illustrates the growing importance of audio effect studies by providing an extensive overview of available datasets and methods.

While tools exist for managing audio datasets, such as `mirdata` [23], they present limitations in the specific context of this thesis. `mirdata` provides standardized loaders for various MIR datasets, promoting reproducibility. However, it is a general-purpose library and lacks a specific focus on the requirements of guitar effect removal research, such as handling paired clean/effected data generation or integrating specific augmentation and normalization pipelines tailored for this task. Furthermore, its reliance on NumPy often necessitates that researchers implement custom data loading and preprocessing logic using frameworks like PyTorch to integrate effectively with deep learning training loops. The toolkit developed in this thesis aims to overcome these limitations by providing native PyTorch integration, focusing specifically on aggregating relevant guitar datasets (including those not currently in `mirdata`), and incorporating configurable on-the-fly normalization and augmentation pipelines designed explicitly for generating training data for effect removal models from clean source audio.

Collectively, the existing literature validates the significance of addressing data diversity, normalization, and augmentation challenges. The methodological choices adopted in this thesis are informed by these prior works, ensuring the proposed toolkit is grounded in the state-of-the-art while addressing critical gaps in current tooling and research practices,

particularly the need for a unified, flexible, and deep-learning-native platform for guitar audio effect data aggregation and generation.

2. Theoretical Background

2.1 Audio Effects

Audio effects play an essential role in contemporary music production and sound design, providing tools that enable musicians and engineers to shape audio signals through modifications in timbre, pitch, dynamics, and spatial characteristics. As highlighted by Reiss and McPherson [24], audio effects are both creative instruments and subjects of technical investigation in digital audio signal processing.

From their analog origins in tube amplifiers and electromechanical devices to today's advanced digital implementations, audio effects have continually evolved alongside technological innovation. Zölzer [25] emphasizes that digital signal processing has significantly enhanced the precision and creative potential of audio effect modeling. The historical context provided by Wilmering [26] further illustrates how interdisciplinary advancements have propelled innovations in this field.

This chapter comprehensively reviews audio effects by examining their historical development, theoretical foundations, and modern digital approaches. Particular emphasis is placed on guitar-oriented audio effects, especially the nonlinear drive effects crucial in shaping guitar tones, which sets the stage for subsequent discussions about guitar-specific music information retrieval (MIR) tasks.

2.1.1 Historical Development of Audio Effects

The evolution of audio effects is marked by a constant interplay between artistic creativity and technological progress. Initially, composers leveraged natural acoustics and architectural spaces to manipulate reverberation and echo, laying the foundation for future artificial reverberation techniques [26].

With the advent of electronic amplification and recording technologies in the twentieth century, intentional sound modification became increasingly prominent. Tube amplifiers introduced desirable nonlinearities, producing warmth and distortion characteristic of certain musical genres [24]. Concurrently, electromechanical innovations such as spring-based delay lines expanded the creative toolbox for sound engineers, underpinning many contemporary audio effects [26].

The transition to digital audio signal processing represented a significant advancement, enabling precise emulation and innovative digital algorithms. Zölzer [25] details how digital audio effects (DAFX) have provided unprecedented parameter control and complexity, significantly influenced by innovations from telecommunications, computer science, and film technology [26].

Overall, the historical trajectory of audio effects demonstrates a progressive convergence of creative vision and technological advancements, directly influencing today's music production practices.

2.1.2 Theoretical Foundations and Categorization of Guitar Audio Effects

To understand audio effects within guitar signal processing, it is crucial to examine their theoretical underpinnings. Digital audio processing models signals as discrete data streams and employs both linear and nonlinear systems. Linear systems typically preserve spectral characteristics, while nonlinear systems—prevalent in guitar effects—introduce harmonically complex distortions through clipping and waveshaping [24, 25].

Four primary categories emerge when analyzing guitar audio effects:

1. **Dynamic Effects**: Compressors, limiters, and gates that manage amplitude dynamics without significantly altering harmonic content [24].
2. **Time-Based Effects**: Delays and reverbs, which add spatial depth and richness by temporal manipulation of the signal [25].
3. **Modulation Effects**: Chorus, flanger, phaser, vibrato, and tremolo, involving modulated signal copies to produce variations in pitch, phase, or amplitude, enriching sound textures [25].
4. **Drive Effects**: Including overdrive, distortion, and fuzz, these nonlinear effects drastically reshape the guitar's sonic identity through waveform clipping, generating harmonically rich textures essential to numerous musical styles. Due to their complexity, they pose significant challenges in tasks such as audio effect removal, highlighting the unique research opportunities in guitar-specific MIR tasks [24, 26].

As seen in the Figure 1, time-based effects like delay introduce discernible repetitions. Modulation effects such as phaser alter the waveform's phase characteristics, often creating a sweeping visual texture. Drive effects, like distortion, typically result in a more compressed and clipped waveform altering the original waveform the most, visually

representing the added harmonic richness and sustain.

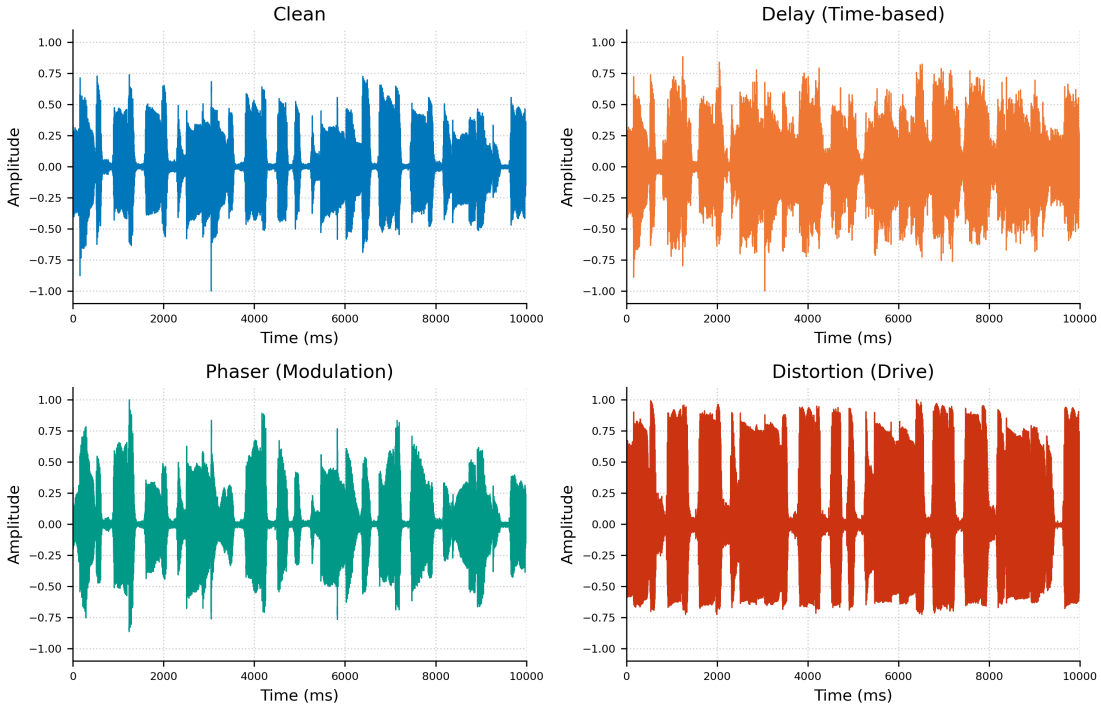

Figure 1. Electric guitar audio effects waveforms

Understanding these categories and their visual manifestations provides essential context for exploring digital modeling techniques and their application within guitar signal processing, especially regarding nonlinear drive effects. It is worth noticing that in this thesis the term distortion will be used to refer to all drive effects, including overdrive, fuzz, and distortion. This is due to the fact that in the context of guitar audio processing, these effects are often used interchangeably and share similar principles of operation, primarily involving nonlinear signal processing techniques to create harmonic richness and sustain.

2.1.3 Digital Modeling Approaches

Digital modeling has revolutionized audio effects, enabling precise replication of traditional analog effects and innovative new processes within digital audio workstations (DAWs) [24, 25].

Traditional Frameworks: JUCE and VST. The JUCE framework [27] is widely adopted for developing cross-platform audio plugins, facilitating integration across various DAWs and formats, thus promoting innovative effect modeling [24].

Accessible Interfaces: Pedalboard. Pedalboard [8], built atop JUCE, offers streamlined

development and prototyping capabilities, significantly lowering barriers to entry for audio effect design.

Neural Network-Based Methods. Recently, neural network techniques have emerged as powerful tools for modeling complex nonlinear effects, particularly drive effects. Methods such as GuitarML [11] and NeuralAmpModeler [12] utilize deep learning to achieve highly accurate representations of analog distortions, addressing limitations inherent in traditional DSP methods.

Together, these modeling techniques form the foundation of modern audio effect processors, highlighting the intersection of traditional DSP practices with innovative machine learning methodologies, particularly relevant in the context of guitar-oriented music information retrieval tasks discussed in the following chapters.

2.2 Music Information Retrieval

Music Information Retrieval (MIR) has seen significant advancements in recent decades, with automatic music transcription emerging as one of its most extensively researched tasks. Early studies laid the groundwork for accurately extracting note onsets, durations, and pitches from guitar recordings [3]. These pioneering efforts not only advanced transcription techniques but also established datasets—such as *GuitarSet* and EGDB—that have become essential resources for both transcription and effect removal research. Notably, the high-quality, annotated data from these datasets provide the clean guitar performance signals that underpin recent deep learning approaches, which in turn have spurred progress in tasks like audio effect removal.

2.2.1 Automatic Music Transcription

In the last decades evolution of automatic music transcription (AMT) for guitar signals was supported by new innovative approaches such as the method proposed by Kehling et al. [3] accompanied by the introduction of new a dataset - IDMT-SMT-Guitar. Their algorithm combined state-of-the-art techniques for onset and offset detection, multipitch estimation, and instrument-specific parameter extraction to achieve high accuracy in both note and technique identification. This early work demonstrated that robust transcription of guitar-based tablature is achievable even in polyphonic settings.

Building on these foundations, approach by Yu-Hua Chen et al. with the release of the *GuitarSet* dataset [1] marked a significant milestone in the field. *GuitarSet* provided

rich, time-aligned annotations—including string and fret positions, chords, and playing styles—that have not only improved transcription models but also offered invaluable training data for related tasks such as effect removal. The availability of such detailed and diverse datasets has been critical in pushing the limits of transcription accuracy.

More recently, the introduction of the EGDB dataset and a multi-loss Transformer model [2] has further advanced the state-of-the-art in guitar transcription. By benchmarking well-known transcription models originally designed for piano and integrating a novel Transformer architecture, this work highlighted the influence of timbre on transcription performance and identified areas where guitar transcription still lags behind its piano counterparts.

Subsequent research has aimed to improve transcription robustness by leveraging real electric guitar tones and audio effects [16]. These studies have shown that incorporating synthetic data—generated from real recordings processed with diverse effects—can significantly enhance the performance of guitar tablature transcription models. Moreover, recent advances in domain adaptation have resulted in high-resolution transcription methods that use score alignment techniques to train models in a zero-shot context [28]. This approach not only bridges the gap between piano and guitar transcription accuracy but also underscores the utility of large-scale, annotated datasets in overcoming data scarcity.

Finally, another work on zero-shot domain adaptation for AMT [21] further demonstrates the potential of adapting pre-trained models—originally developed for piano transcription—to guitar recordings without requiring additional labeled data. By aggregating predictions from pitch-shifted inputs, this method consistently improves transcription performance under various domain shifts.

Together, these contributions—from early algorithmic innovations to state-of-the-art deep learning techniques—have established a robust framework for automatic music transcription. Moreover, the same datasets and methodologies developed for transcription are now instrumental in addressing the inverse problem of audio effect removal, highlighting the interdependent evolution of these MIR tasks.

2.2.2 Audio Effect Removal

The evolution of audio effect removal techniques mirrors the broader development in MIR—from early, model-based methods to current deep learning approaches that benefit from large-scale transcription datasets and advanced music source separation models. Early efforts focused on inverting simpler audio effects, such as dynamic range compres-

sion, using traditional signal processing techniques. For example, Lachaise and Daudet [29] estimated the additional side information needed to exactly or approximately invert a compressor, achieving nearly lossless reconstruction in favorable conditions. Later, Gorlow and Reiss [30] developed a model-based inversion method that exploited known compression parameters to recover the original (dry) signal with high accuracy and low computational complexity. Although these methods address relatively simpler non-linear processes compared to guitar distortion, they laid the groundwork for subsequent effect removal research.

The availability of high-quality guitar transcription datasets—such as *GuitarSet* [1] and EGDB [2]—has been instrumental in advancing effect removal. These datasets, originally developed for automatic music transcription, together with digital audio effect modeling provide clean reference as well as processed audio signals that serve as training data for supervised inversion models. In this way, the transcription community's efforts indirectly support effect removal by supplying the reliable ground truth needed to learn complex non-linear mappings.

Parallel to these developments, sparsity-based methods have been explored for audio declipping—a task conceptually similar to effect removal. Kitic et al. [31] proposed a flexible non-convex approach that compares sparse synthesis and analysis regularization for recovering signals from clipped audio. Although such methods work well for speech, they assume a simpler distortion model than the highly non-linear, memory-dependent drive effects found in electric guitar recordings.

Advances in virtual analog modeling further extended traditional inversion techniques. Bernardini et al. [32] investigated the inversion of analog audio circuits using Wave Digital Filter theory, exploiting circuit nullors to approximate the input signal from the output. However, the inherent complexity of real-world non-linear systems—especially those with memory effects—often limits these approaches.

The advent of deep learning has dramatically transformed effect removal. Early applications of deep filtering for speech declipping [33] demonstrated that neural networks could extract and reconstruct signals in the short-time Fourier transform domain with improved performance over conventional methods. Building on these ideas, Imort et al. [9] leveraged music source separation architectures—such as Demucs, Wave-U-Net, and UMX—to remove distortion effects from guitar recordings, achieving high-quality recovery when the distorted (wet) signal mixes with the original dry signal.

Complementary to inversion methods, accurate recognition and parameter estimation

of guitar effects is critical. Works by Jürgens et al. [34] and Comunità et al. [35] utilize convolutional neural networks to classify and estimate effect parameters, providing valuable priors that can enhance effect removal performance.

Recent research is moving toward more general-purpose solutions capable of handling multiple effects simultaneously. Rice et al. [13] introduced RemFX, which dynamically constructs a graph of effect-specific removal models using audio effect classification. Tanaka et al. [36] proposed APPLADE, integrating a deep neural network within an optimization framework to robustly reverse audio distortions despite mismatches between training and test data. Other promising approaches include blind extraction methods for guitar effects based on hybrid Transformer architectures [18] and two-stage methodologies that first purify the Mel-spectrogram and then use a neural vocoder to reconstruct a high-fidelity dry signal from distorted guitar recordings [37]. Diffusion-based models have also been explored for unsupervised estimation and inversion of unknown non-linear distortions [38]. In addition, work on audio effect chain estimation [17] aims to recover not only the dry signal but also the complete chain of applied effects, while real-time capable approaches like DDD [39] demonstrate low-response-time declipping with adversarial training.

In summary, effect removal research has evolved from traditional compression inversion methods to sophisticated deep learning frameworks that exploit high-quality transcription datasets and state-of-the-art source separation models. Current approaches increasingly integrate effect recognition, general-purpose removal, chain estimation, and multi-stage processing to tackle the complex non-linear distortions typical in guitar recordings—paving the way for improved post-production, remixing, and automated audio editing applications.

Despite significant progress, several challenges remain. In the context of effect removal, the scarcity of large-scale, high-quality training data as noted by many researchers [40, 4, 5] limits the generalization of deep learning models, as often state-of-the-art systems rely on large, proprietary datasets [37]. Similarly, while datasets such as *GuitarSet* and EGDB have propelled advances in guitar transcription, further expansion in data diversity is needed to address the variability present in real-world recordings.

2.3 Datasets

Transitioning from the challenges outlined above, it is essential to assess the available guitar datasets with an eye toward their suitability for effect removal tasks. Many existing datasets were originally designed for transcription and other MIR tasks, which impacts their direct transferability to effect removal. For instance, while datasets such as unnamed

dataset by Schmitz and Embrechts [41], EGFxSet [42], IDMT-SMT-Audio-Effects [43] and Guitar FX DIST [35] offer aligned clean and rendered guitar audio, they are less appropriate for effect removal task due to limitations such as short duration of clean audio as well as lack of control over effect applied to rendered audio. Similarly, resources like AudioSet [44], GuitarDuets [45], GAPS [46], SignalTrain [47], DadaGP [48], and GuitarSoloDetection [49] are constrained by factors including the exclusive availability of annotations, need to render the data or reliance on purely synthetic sound without timbral diversity.

In contrast, several datasets have emerged as robust baselines for guitar effect removal research. Key datasets utilized and aggregated in this thesis include IDMT-SMT-Guitar [3], GuitarSet [1], EGDB [2], and Guitar-TECHS [5]. These offer clean and sufficiently long recordings that facilitate both transcription and effect removal tasks. A summary of their primary characteristics relevant to this work is presented in Table 1.

Table 1. Primary Guitar Datasets

Dataset	**Main Task(s)**	**Data Composition**	**Musical Texture**	**Duration**
IDMT-SMT-Guitar	Transcription	Electric & Acoustic	Mono- & Polyphonic	5.6 h
GuitarSet	Transcription	Acoustic	Polyphonic	3 h
EGDB	Transcription	Electric	Polyphonic	2 h
Guitar-TECHS	Timbre transfer, performance gen., transcription	Electric	Mono- & Polyphonic	5 h

The most recent contribution, Guitar-TECHS, further introduces a diverse range of guitar techniques and timbral variations that hold promise for advancing future research in effect removal.

2.3.1 IDMT-SMT-Guitar Dataset

The IDMT-SMT-Guitar dataset was originally introduced by Kehling et al. [3] as part of a novel approach for automatic tablature transcription of electric guitar recordings. The dataset is divided into several subsets tailored for various transcription tasks. For example, one subset contains single-note and chord recordings with detailed annotations on parameters such as pitch, string number, plucking style, and expression. Additional subsets include a series of guitar licks—both monophonic and polyphonic—as well as short musical pieces recorded under different playing conditions.

Despite its comprehensive annotation, the dataset's limited size (e.g., only 17 unique guitar licks for transcription purposes) has been noted as a constraining factor in achieving robust generalization in deep learning models [2, 1]. Nevertheless, the fine-grained annotations have also made the dataset an attractive resource beyond transcription. For instance, Wright et al. [50] employed IDMT-SMT-Guitar (in conjunction with its bass counterpart) to train neural network models for real-time guitar amplifier emulation, while Comunità et al. [51] and Hinrichs et al. [18] leveraged portions of the dataset for modeling and blind extraction of guitar effects.

In further research, the dataset's utility has been extended to novel tasks. Švento et al. [38] used a small portion of the IDMT-SMT-Guitar dataset as part of a diffusion-based approach to restore nonlinearly distorted audio. Take et al. [17] also integrated this dataset into their pipeline for audio effect chain estimation and dry signal recovery. Moreover, its role as a benchmark for guitar transcription has spurred the development of augmented and synthetic datasets—such as SynthTab [4] and more recent collections [10]—to overcome the limitations in timbral and expressive diversity inherent to the original IDMT-SMT-Guitar dataset.

Overall, while the IDMT-SMT-Guitar dataset has been foundational in guitar-related Music Information Retrieval (MIR) research, its relatively narrow scope has motivated further expansion and innovation in dataset design for both transcription and effect removal tasks.

2.3.2 GuitarSet Dataset

GuitarSet, introduced by Xi et al. [1], is a seminal dataset for guitar transcription. It provides 360 excerpts of acoustic guitar recordings with rich annotations including string and fret positions, chords, beats, and downbeats. The dataset was designed to overcome limitations of earlier collections by capturing detailed note-level information and enabling the exploration of additional research directions such as stroke analysis and harmony segmentation.

Following its introduction, GuitarSet quickly became a de-facto standard in guitar Music Information Retrieval (MIR). For instance, Bittner et al. [23] incorporated GuitarSet into the `mirdata` library, addressing reproducibility issues by providing standardized data loaders and validation tools. This effort helped ensure consistent usage of GuitarSet in subsequent research.

Sarmento et al. [48] further highlighted GuitarSet's role by using it as a reference point in their symbolic music generation dataset, DadaGP, while noting that the dataset's focus

on acoustic recordings and its limited timbral diversity could restrict its utility for broader musical applications.

Building on this, Chen et al. [2] contrasted GuitarSet with their newly introduced EGDB, pointing out that GuitarSet offers only 30 unique comping and solo tracks recorded in a single timbre. This limitation has motivated the creation of datasets that capture a broader range of electric guitar tones.

In the same vein, Pedroza et al. [42] emphasized that while GuitarSet's meticulous annotations are valuable for transcription, the absence of effect processing renders it less suitable for tasks that involve modeling real-world audio effects. Later, Rice et al. [13] demonstrated that augmenting GuitarSet using Pedalboard [8] audio effects can extend its applicability to general purpose audio effect removal tasks.

Further work by Chen et al. [40] employed GAN-based techniques for clean-to-rendered guitar tone transformation, using GuitarSet as a baseline for comparing paired clean and effected audio. Similarly, Hinrichs et al. [18] adapted GuitarSet by segmenting and augmenting its recordings to form GuitarSetEQ and GuitarSetVFX, which served as the foundation for their work on blind extraction of guitar effects.

More recent studies have integrated GuitarSet into complex processing pipelines. Pedroza and Abreu [16] combined GuitarSet with synthetic data to improve robustness in guitar tablature transcription, and Riley et al. [28] applied domain adaptation techniques to achieve high-resolution transcription performance on GuitarSet. Additionally, the emergence of larger datasets such as Guitar-TECHS [5] reflects ongoing efforts to address GuitarSet's limitations by incorporating diverse timbral and recording conditions.

Together, these works underscore the pivotal role of GuitarSet as a benchmark in guitar MIR research, while also highlighting its limitations in terms of timbral diversity and effect modeling. This has motivated further dataset development and methodological innovations aimed at broadening the scope of guitar-related tasks, including robust transcription and effect removal.

2.3.3 EGDB Dataset

The EGDB dataset was introduced by Yu-Hua Chen et al. [2] to overcome limitations inherent in earlier guitar transcription datasets. Unlike datasets based on acoustic recordings such as GuitarSet, EGDB focuses on electric guitar performances. By capturing direct input (DI) recordings with a hexaphonic pickup and re-rendering these recordings using

different amplifier settings, EGDB provides transcriptions for 240 tablatures rendered in six distinct timbres, totaling approximately 118 minutes of audio. This multi-timbre design addresses the variability of electric guitar sounds and helps mitigate the historical gap between guitar and piano transcription performance.

Building on this foundation, subsequent works have further explored the potential of EGDB. Pedroza et al. [42] introduced the EGFxSet dataset, which extends EGDB by processing its clean recordings through real effects hardware. Although EGFxSet improves effect realism by incorporating parameters from real-world devices, it is limited to monophonic tones, highlighting an ongoing trade-off between effect authenticity and musical complexity.

In parallel, Chen et al. [40] leveraged EGDB in a GAN-based framework aimed at clean-to-rendered guitar tone transformation. Their approach, which benefits from unpaired data across multiple sources, demonstrated improved modeling for both low-gain and high-gain amplifier effects—underscoring the value of the timbral diversity provided by EGDB.

The dataset's impact extends into effect removal research as well. Lee et al. [37] utilized EGDB as the source of dry signals for training a two-stage distortion recovery model. Their method, which first purifies the audio signal in the Mel-spectrogram domain and then reconstructs it using a neural vocoder, illustrates how EGDB can serve as a robust reference for recovering original guitar tones from effected signals.

Addressing challenges related to limited data, Švento et al. [38] pre-trained a diffusion model on EGDB's DI recordings to combat overfitting, thereby highlighting EGDB's importance as a source of clean electric guitar data for modeling nonlinear distortion.

Finally, Wright et al. [10] incorporated EGDB into their Open-Amp framework for synthetic data augmentation. By using EGDB as input to generate a wide range of augmented guitar effects, they demonstrated improved generalization of guitar effects models in large-scale training scenarios.

Collectively, these studies illustrate that EGDB has become a critical benchmark for both electric guitar transcription and effect processing. Its emphasis on multi-timbre recordings and systematic re-rendering has provided a rich resource for advancing state-of-the-art techniques in both transcription and effect removal.

2.3.4 Guitar-TECHS Dataset

The Guitar-TECHS dataset [5] is a recently introduced, comprehensive electric guitar dataset designed to address the limitations of previous collections. Unlike earlier datasets that focus primarily on acoustic guitar—such as GuitarSet [1]—Guitar-TECHS offers multi-perspective recordings by capturing audio via four distinct setups: direct input, a miked amplifier, an egocentric (player) microphone, and an exocentric (listener) microphone. This rich diversity, combined with precise MIDI annotations of guitar techniques, musical excerpts, chords, and scales, provides over five hours of combined monophonic and polyphonic electric guitar content. Such detailed annotation and varied recording conditions are particularly valuable for advancing tasks in guitar tablature transcription (GTT) and other guitar-related machine listening applications.

Due to its novelty, the number of studies citing Guitar-TECHS remains limited. However, early investigations already point to several important insights. Preliminary experiments have shown that incorporating Guitar-TECHS into training pipelines for GTT can enhance model robustness—improving metrics such as tablature disambiguation and multi-pitch estimation. Additionally, the dataset's multi-perspective design is recognized as highly promising for emerging applications in augmented and virtual reality, where capturing both the performer's and listener's perspectives can significantly enrich user experience. The rich annotations provided by Guitar-TECHS are also expected to facilitate more nuanced analyses of guitar timbre and playing techniques, which may prove crucial in future research on effect removal and cross-view learning.

In summary, Guitar-TECHS fills an important gap by offering increased acoustic diversity and multi-modal audio captures, making it a valuable benchmark for future data-driven guitar research.

2.4 Normalization

Normalization is a critical preprocessing step in Music Information Retrieval (MIR) systems, particularly when addressing complex tasks such as audio effect removal. The quality and consistency of normalization methods directly affect the performance and generalization of deep learning models. In this chapter, an overview of normalization techniques used in MIR is provided, discussing early approaches, the widespread application of internal normalization methods such as batch normalization, and the limitations of traditional peak normalization compared to perceptually motivated strategies like loudness normalization.

Early MIR research often relied on simple amplitude scaling methods such as peak normalization. While peak normalization adjusts the maximum amplitude of signals to a common reference (e.g., a fixed dBFS level), it does not account for perceptual loudness differences. As pointed out by [52] in the context of soundscape synthesis, two sounds normalized by their peak values may still be perceived as having different loudness levels. This limitation becomes especially critical in tasks like effect removal, where maintaining the dynamic and timbral nuances of the audio is essential.

In recent years, deep neural network (DNN) architectures have predominantly incorporated batch normalization within their layers. For instance, several works on audio declipping [33] and virtual analog modeling [53] utilize batch normalization to stabilize and accelerate training. Similarly, studies in industrial sound analysis demonstrate the use of adaptive normalization methods that mitigate domain shift [54]. However, while batch normalization is effective in an internal network context, its role as a preprocessing normalization step has not been thoroughly examined. There remains an open question as to whether omitting external normalization in favor of solely relying on internal batch normalization might compromise the preservation of essential musical dynamics.

Despite its widespread use, peak normalization suffers from fundamental shortcomings. Works such as [55] continue to employ peak normalization (e.g., applying a -12 dBFS reference), yet this method fails to ensure that the perceived loudness of different signals is consistent. Perceptually, equalizing loudness rather than merely matching peak amplitudes is more aligned with human auditory perception. The `pyloudnorm` library, described in [7], implements loudness normalization based on the ITU-R BS.1770 standard (measuring in LUFS), providing a more robust and perceptually relevant normalization. This approach has demonstrated improved consistency across various audio contents and is particularly beneficial in tasks where the dynamic range and timbral details are paramount, such as in the removal of complex distortion effects.

In effect removal tasks, the preservation of subtle audio characteristics is essential for recovering a clean signal from processed or distorted recordings. While many studies on effect removal integrate batch normalization within their DNN frameworks (e.g., [33, 56]), there is a notable lack of systematic research on how external normalization techniques influence the performance of these models. Recent works in automatic music mixing and amplifier modeling have shown that proper normalization can alleviate data scarcity and domain shift issues [57, 55]. However, a comprehensive evaluation contrasting the benefits of perceptual loudness normalization with traditional peak and batch normalization is still needed. Given its ability to provide uniform perceived loudness, loudness normalization presents a promising direction for enhancing the robustness of effect removal systems.

The literature reveals that while internal normalization via batch normalization is a staple in modern deep learning architectures, the impact of external normalization strategies has not been sufficiently explored in MIR. The limitations of peak normalization and the perceptual advantages of loudness normalization suggest that future research should systematically investigate how these preprocessing steps affect the generalization and performance of effect removal models.

2.5 Audio Effects as Augmentations

Audio effects as a form augmentation of audio data in Music Information Retrieval (MIR) has undergone significant evolution, particularly for the task of audio effect removal. Early research focused on the nonlinear modeling of guitar signal chains, which established foundational methods for accurately emulating hardware effects in real time [58]. Such pioneering work underscored the importance of capturing nonlinear behaviors to support downstream processing tasks, including the removal of complex audio effects.

2.5.1 Baseline Effect Application Methods

A variety of tools have been developed to facilitate audio data augmentation in MIR. Notably, the `Pedalboard` library has gained considerable popularity due to its accessibility and ease-of-use [18, 37, 17]. Pedalboard offers a range of basic audio effects (e.g., distortion, delay, chorus, and reverb), making it a useful tool for generating baseline augmented datasets. However, its implementations tend to be simplified and do not fully capture the tonal diversity of real-world hardware. As such, while Pedalboard is effective for initial evaluations and benchmarking, its limited realism can constrain the generalization of models trained on such augmented data.

2.5.2 Advanced Effect Application with VST Plugin Effects

To overcome the limitations of simplified tools, several studies have explored the use of commercial-grade VST plugin effects [18, 37, 10]. VST plugins are capable of emulating the intricate behaviors of analog devices by incorporating detailed circuit and acoustic modeling. Despite their potential to yield more realistic augmented data, VST plugins present several challenges. High-quality plugins are often commercial products, restricting their accessibility, and many are optimized for macOS rather than Linux or high-performance computing (HPC) environments. Furthermore, their effective use typically requires a deep understanding of the specific plugin's characteristics, which can be a barrier for researchers.

2.5.3 Neural Network Models for Effect Application

Recent advances in neural network modeling offer a promising alternative for audio effects augmentation. Open-source frameworks—such as GuitarML and Neural Amp Modeler—have emerged as compelling solutions by leveraging large, community-curated datasets to model a wide variety of amplifier and effect characteristics [10]. These neural approaches enable the synthesis of highly diverse and realistic augmented data, which is critical for training robust models for effect removal. In addition to their scalability and adaptability, these methods facilitate real-time emulation and provide a flexible framework for generating paired clean and processed audio samples. Their advantages are particularly notable when compared with the more constrained capabilities of traditional methods [10].

The literature indicates a clear trajectory from baseline augmentation tools such as `Pedalboard` [37, 17], to inconvinient VST plugin based processing and finally toward more advanced neural network-based frameworks [10]. While traditional tools provide valuable benchmarks, neural approaches promise greater realism and diversity, which are crucial for enhancing the generalization of effect removal systems. Future research should continue to address challenges related to preserving the dynamic and timbral nuances of original recordings during augmentation, and explore integrated normalization strategies to further improve model robustness [18].

3. Technical Implementation

3.1 Functional Requirements

Building on the comprehensive theoretical framework that explores the intricate dynamics of guitar audio effects, effect removal challenges, and state-of-the-art normalization and augmentation methods, the technical implementation outlined in this chapter transforms these insights into a practical solution. By adopting an agile methodology structured around clearly defined epics (see Table 2) and user stories (see Table 3), the development process systematically addresses the core challenges identified in the literature. This approach enables the integration of diverse guitar datasets, the application of perceptually robust normalization techniques, and the deployment of advanced augmentation strategies, thereby bridging the gap between theoretical innovation and real-world application in audio effect removal.

Table 2. Epics

Epic ID	**As a <type of user>**	**I want to <perform some task>**	**so that I can <achieve some goal>**
1	Researcher	Download guitar datasets	Use them for my research
2	Researcher	Specify the type of data from a dataset to use	Use data best suitable for my research
3	Researcher	Normalize datasets	Improve the model training process
4	Researcher	Augment datasets with effects	Train models for effect removal

Table 3. User Stories

User Story ID	**User Story Priority**	**As a <type of user>**	**I want to <perform some task>**	**so that I can <achieve some goal>**
1.1	High	Researcher	Download IDMT-SMT-Guitar	Use its data for my research
1.2	High	Researcher	Download GuitarSet	Use its data for my research

Continues...

Table 3 – *Continues...*

User Story ID	**User Story Priority**	**As a <type of user>**	**I want to <perform some task>**	**so that I can <achieve some goal>**
1.3	High	Researcher	Download EGDB	Use its data for my research
1.4	High	Researcher	Download Guitar-TECHS	Use its data for my research
2.1	High	Researcher	Specify several datasets	Use more data for my research
2.2	High	Researcher	Specify acoustic and/or electric guitar data from a dataset to use	Use data best suitable for my research
2.3	Low	Researcher	Specify mono- and/or polyphonic data from a dataset to use	Use data best suitable for my research
2.4	Low	Researcher	Specify datasets used for pretraining, training, validation, and testing	Use configuration best suitable for my research and enable cross-validation
3.1	High	Researcher	Use loudness normalization	Improve the model training process
3.2	Medium	Researcher	Use peak normalization	Improve the model training process
4.1	Medium	Researcher	Augment datasets with Pedalbord	Train models using synthetic effects
4.2	High	Researcher	Augment datasets with GuitarML	Train models using realistic effects

The overall data processing pipeline envisioned and implemented in this toolkit is depicted in Figure 2. Initially, various established guitar datasets such as IDMT-SMT-Guitar, GuitarSet, EGDB, and Guitar-TECHS, among others, are aggregated. This aggregated collection then undergoes a normalization stage, where techniques like peak or loudness normalization can be applied. Subsequently, data augmentation is performed, offering options for both baseline (e.g., simple waveshaping) and advanced (e.g., neural network-based) effect simulation. Finally, this prepared data is used to train effect removal models, such as the RemFX framework utilized in the experiments.

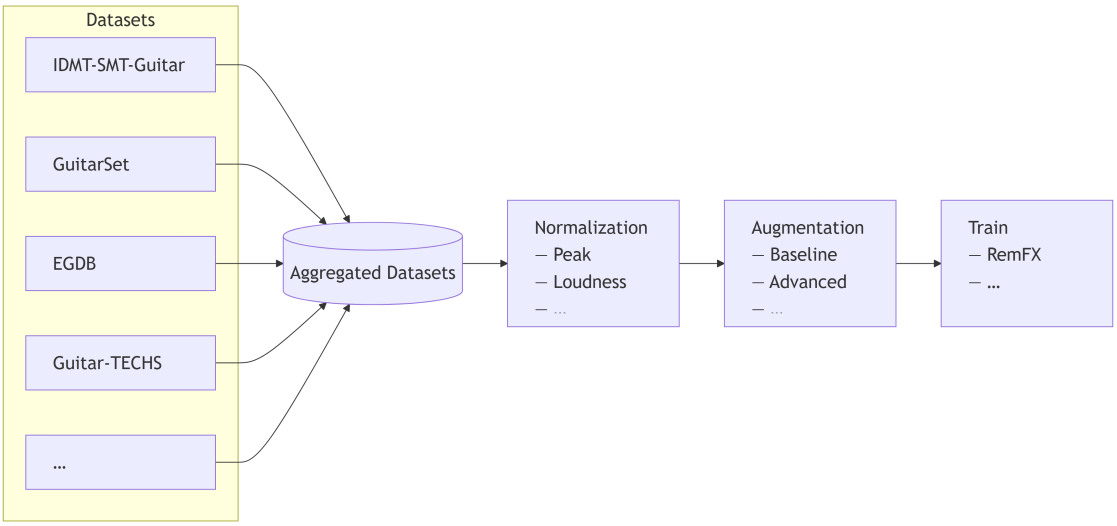

Figure 2. Data aggregation and processing pipeline

3.2 Frameworks

To build a robust and reproducible toolkit for aggregating and generating real-world guitar audio data, current implementation leverages three core frameworks: PyTorch [59], Hydra [60], and PyTorch Lightning [61]. Each of these frameworks addresses distinct challenges in deep learning and software configuration, ensuring that developed system remains efficient, reproducible, and scalable.

3.2.1 PyTorch

PyTorch serves as the backbone for deep learning components due to its dynamic, imperative programming style and seamless GPU acceleration, which greatly facilitate rapid prototyping and debugging [59]. Unlike libraries such as `mirdata` [23], which are built on NumPy and often force researchers to re-implement data loaders repeatedly, PyTorch offers native support through its `Dataset` and `DataLoader` abstractions. This not only streamlines data preprocessing but also enforces consistency and reproducibility in handling diverse guitar audio datasets.

3.2.2 Hydra and hydra-zen

Hydra is a powerful framework that enables the dynamic composition of hierarchical configurations, ensuring that every experimental run is fully reproducible by automatically recording its complete configuration state [60]. In developed toolkit, Hydra is employed

to manage configurations for datasets which enables high degree of reproducibility. In addition, the hydra-zen [62] extension offers a Python-centric workflow for automatically generating and validating YAML configurations. Although hydra-zen has not yet become a mainstream tool, its ability to significantly reduce the manual overhead and technical debt associated with configuration management makes it a promising extension to Hydra.

3.2.3 PyTorch Lightning

PyTorch Lightning abstracts much of the engineering boilerplate inherent in standard PyTorch training loops, thereby allowing to focus on developing and refinin models [61]. Lightning enforces best practices by decoupling scientific code from engineering concerns, which results in cleaner, more maintainable code. Furthermore, its built-in support for distributed training strategies simplifies scaling experiments across multiple GPUs. This streamlined approach not only accelerates development but also enhances reproducibility and consistency across different hardware configurations.

3.3 Implementation

This section details the technical realization of the toolkit's core components, following a process that encompasses data aggregation and preprocessing, audio segmentation, normalization, augmentation, dataset loading, and optional rendering. The implementation prioritizes efficiency, reproducibility, and flexibility for research purposes. Figure 3 illustrates the overall architecture and data flow of these components, from initial data acquisition to the final data representation used for model training or experimentation. This includes components for downloading, extracting and concatenating datasets as well as segmenting audio files, generating metadata, applying normalization and augmentation, and loading datasets for training. The toolkit is designed to be modular and extensible, allowing researchers to adapt it to their specific needs while maintaining a consistent interface for data handling.

3.3.1 Data Aggregation and Preprocessing

The initial stage focuses on acquiring raw audio data from diverse sources and preparing it for subsequent processing. This involves downloading datasets and then extracting, categorizing, and concatenating the audio files.

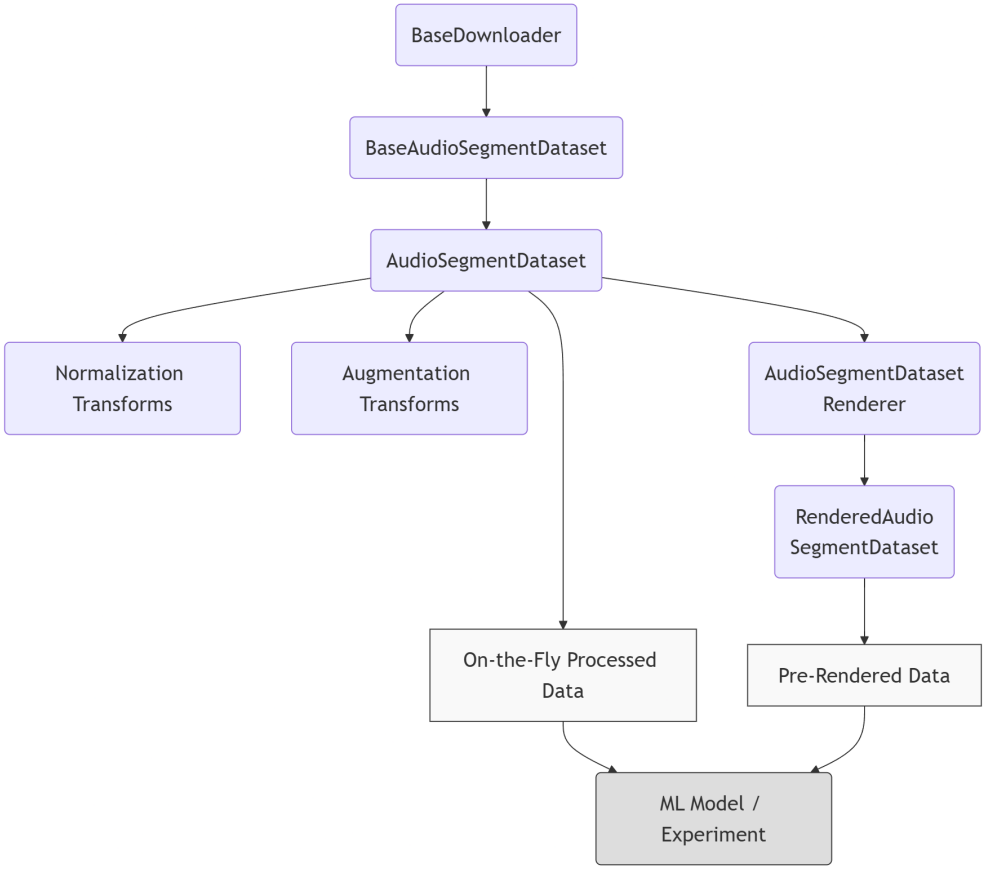

Figure 3. Architecture and data flow of the toolkit components

Downloading

To ensure maintainability and ease of adding future datasets, downloader functionalities are built upon an abstract `BaseDownloader` class. Specific downloaders inheriting from this base were implemented for the target datasets: IDMT-SMT-Guitar [3], GuitarSet [1], EGDB [2], and Guitar-TECHS [5].

The download process adapts to the hosting specifics of each dataset. IDMT-SMT-Guitar and GuitarSet are hosted on Zenodo as single ZIP archives, which are downloaded directly using streaming requests. EGDB, hosted on Google Drive, presents challenges due to API limitations for unauthenticated bulk downloads. The implementation uses the `gdown` library with a workaround that scrapes the public folder view to obtain individual file IDs, allowing download without requiring user credentials, although this method is sensitive to changes in Google Drive's web interface. Downloaded EGDB files are then packaged locally into a single ZIP archive for consistent handling. Guitar-TECHS, also from Zenodo, is provided as a single archive containing nested ZIP files, requiring specialized handling during the extraction phase. Progress bars are integrated for monitoring large downloads.

Extraction and Concatenation

Following the download, audio data is extracted and organized. The extraction logic, managed within the `BaseDownloader` structure, adapts to the varying archive structures. GuitarSet and the repackaged EGDB archive contain a flat structure of audio files, corresponding directly to acoustic polyphonic and electric polyphonic categories, respectively.

IDMT-SMT-Guitar and Guitar-TECHS contain multiple data types within their archives. The `BaseDownloader` allows specifying filtering conditions (based on file paths and keywords) to categorize and extract specific data subsets. For IDMT-SMT-Guitar, filters identify electric monophonic, electric polyphonic (using keywords like `Chords`, `LickX`, and path components), and acoustic polyphonic data. For Guitar-TECHS, filters target electric monophonic and electric polyphonic data (using keywords like `directinput`, `chords`, `music`), while also handling the nested ZIP structure inherent to its distribution format. Based on specified characteristics of these datasets `BaseDownloader` is designed in such a way as to provide common interface to handle described issues.

A key preprocessing step is the concatenation of all extracted audio files belonging to the same category (e.g., electric monophonic) into a single large `.wav` file. This strategy optimizes storage space and subsequent data loading efficiency. Performance evaluations were conducted to determine the most efficient concatenation method. Initial tests comparing various libraries (`pydub`, `librosa`, `soundfile`, `pedalboard`) with a naive sequential approach identified `pydub` as significantly faster due to its direct byte-level operations. Further experiments compared different merging algorithms using `pydub` and `soundfile` on the GuitarSet dataset (approx. 3 hours of audio). A heap-based optimal merge pattern strategy, which iteratively merges the smallest segments first, demonstrated superior performance compared to naive, merge-sort-inspired, and quick-sort-inspired approaches, drastically reducing concatenation time. The results comparing `pydub` and `soundfile` with different strategies on the 3 hours of GuitarSet data are summarized in Table 4.

Table 4. Audio Concatenation Efficiency Comparison

Strategy	**`soundfile` time (seconds)**	**`pydub` time (seconds)**
Naive	149.28	65.85
Merge	38.09	4.74
Quick	16.19	4.48
Heap	4.56	2.27

Based on these results, the heap-based optimal merge pattern using `pydub` was adopted for all concatenation tasks within the toolkit.

3.3.2 Audio Segmentation and Metadata Generation

To efficiently handle the large concatenated audio files during training, a segmentation module divides them into smaller, manageable chunks. This process avoids saving numerous small audio files by instead generating metadata.

The segmenter iterates through each concatenated file, extracting segments of a specified duration. This segment length is configurable to suit different experimental needs. During this process, silent or near-silent segments are identified and discarded using a method inspired by prior work [63]. This involves analyzing 1-second sub-chunks within each potential segment; if the volume of a sub-chunk falls below a configurable silence threshold (dBFS), it's marked silent. A segment is only considered valid if the proportion of non-silent sub-chunks exceeds a configurable minimum percentage. This filtering enhances robustness and prevents numerical issues associated with normalizing silence.

For each valid segment, its starting frame offset and duration (number of frames) within the parent concatenated file are recorded. This metadata is stored compactly in PyTorch's `.pt` format, enabling rapid lookup and loading of specific audio segments later without needing to read the entire multi-hour file.

3.3.3 Normalization Implementation

Audio normalization is applied as a transformation step during data loading to ensure consistent levels for model training. Two primary methods are provided, both configurable:

- **Peak Normalization:** Scales the waveform so its maximum absolute amplitude reaches a target level, typically set to -1.0 dBFS by default.
- **Loudness Normalization:** Measures the integrated loudness (LUFS) of the audio segment using the `pyloudnorm` library (implementing ITU-R BS.1770) and scales it to match a target loudness, defaulting to -32.0 LUFS. This provides better perceptual consistency across diverse audio content.

The choice of method and target levels can be adjusted based on research requirements.

3.3.4 Augmentation Implementation

To simulate real-world conditions and improve model generalization for tasks like effect removal, audio augmentation is applied to the clean (dry) audio segments. Two distinct augmentation strategies are available:

1. **Baseline Augmentation:** Implements a hyperbolic tangent (`tanh`) function applied after a random gain multiplication. This efficiently simulates soft-clipping distortion, a common guitar effect. The gain range is configurable. This method is based on `Distortion` effect provided by `Pedalboard` library [8] though since the later operates using `numpy` arrays, a reimplementation using `pytorch` tensors was necessary to ensure compatibility and effectiveness.
2. **Advanced Augmentation:** Leverages pre-trained neural network models from the Open-Amp framework [10]. This uses a Time-Conditioned Network (TCN) capable of emulating a wide variety of guitar amplifiers and distortion pedals (382 distinct models in total, ranging from subtle amp coloration to heavy distortion). The module cycles through these models during processing, applying realistic and diverse effects. While more computationally demanding, this provides high-fidelity augmentation closely mimicking real-world signal chains.

The selection between baseline and advanced augmentation, along with their respective parameters such as range for gain of random tanh distortion, is configurable.

3.3.5 Dataset Loading and Configuration

The toolkit provides a flexible data loading system built on PyTorch's `Dataset` and `DataLoader` abstractions. A `BaseAudioSegmentDataset` class handles the low-level loading: given a path to a concatenated audio file and its metadata file, it retrieves the metadata for a specific index, loads only the required audio segment from the large file, and performs necessary channel selection and resampling.

The primary interface for users is the `AudioSegmentDataset` class. This class aggregates multiple `BaseAudioSegmentDataset` instances based on user-specified keywords. Available keywords correspond to the extracted categories: `acoustic`, `electric`, `monophonic`, `polyphonic`, and dataset identifiers like `idmt-smt-guitar`, `guitarset`, `egdb`, `guitar-techs`. Researchers can combine these keywords (e.g., `['electric', 'polyphonic', 'guitar-techs']`) to precisely define the data mixture for an experiment. This class also manages reproducible train/validation/test

splitting using a fixed random seed.

Crucially, all normalization and augmentation transformations are applied on-the-fly within the `AudioSegmentDataset`. When a clean (dry) segment is loaded, the selected augmentation is applied to create the wet version. Subsequently, the selected normalization is applied independently to both the wet and the original dry segments before they are returned. This on-the-fly approach allows leveraging any available clean guitar audio, as it does not depend on pre-existing paired wet/dry datasets, significantly expanding the potential training data pool.

3.3.6 Dataset Rendering

While on-the-fly processing offers maximum flexibility, the computational cost of repetative augmentation, especially the neural network-based approach, can be significant. To address this, an optional rendering pipeline is provided. The `AudioSegmentDatasetRenderer` module takes a configured `AudioSegmentDataset` (specifying keywords, split, augmentation, normalization) and iterates through it, saving each generated wet/dry segment pair as separate `.wav` files to disk. These are organized in a structured directory hierarchy based on the configuration and segment index.

A corresponding `RenderedAudioSegmentDataset` class allows loading these pre-rendered datasets efficiently. This is beneficial for computationally intensive experiments, sharing standardized datasets, or scenarios where augmentation parameters are fixed.

In summary, the technical implementation provides a robust and efficient toolkit for managing real-world guitar audio data. It features automated aggregation and optimized preprocessing, configurable segmentation with silence removal, flexible on-the-fly normalization and augmentation using both baseline and advanced techniques, and a powerful dataset loading system allowing fine-grained data selection and splitting. An optional rendering pipeline further enhances efficiency for specific use cases. The components are designed for research flexibility and reproducibility. While validated for compatibility with configuration frameworks like Hydra [60] and execution frameworks like PyTorch Lightning [61], these are not strict dependencies, ensuring broader usability within common research workflows. Furthermore native compatibility with `pytorch` and focus on clean guitar audio prove as significant improvement over `mirdata` [23], which requires re-implementation of data loaders and processing steps such as normalization and augmentation and do not account for the specific needs of guitar effect removal task.

4. Experiments

To validate the performance and robustness of the implemented toolkit for guitar distortion removal, a series of experiments were designed. These experiments leverage the RemFx framework [13], chosen for its robust approach to audio effect removal and its convenient integration with PyTorch Lightning and Hydra. This integration facilitated the management of diverse experimental setups and ensured compatibility with frameworks mentioned. For all training runs, the Hybrid Demucs model architecture, a state-of-the-art model originally developed for instrument separation and adapted within RemFx for distortion removal, was utilized.

The experimental design focused on evaluating various configurations of the data aggregation, normalization, and augmentation pipeline. A total of 14 distinct Hydra configurations were defined to systematically explore the impact of:

- Different data subsets (e.g., acoustic, electric, combined, monophonic, polyphonic).
- Normalization strategies (peak-based vs. loudness-based at -23 LUFS and -32 LUFS).
- Augmentation methods (baseline `tanh` distortion vs. advanced NN-based distortion).

For consistency and to prevent data leakage, the underlying clean audio data used for generating training, validation, and test segments was strictly separated across all configurations. Training was conducted for 8000 steps per epoch, while validation and testing each used 1000 steps per epoch, with early stopping employed to prevent overfitting.

The primary evaluation metrics were chosen to assess both signal fidelity and spectral similarity, alongside the improvement achieved by the models. These include:

- **Scale-Invariant Signal-to-Distortion Ratio (SI-SDR)** [14]: Reported in decibels (dB), where higher values (↑) indicate better reconstruction quality relative to the clean source.
- **Multi-Resolution Short-Time Fourier Transform (MRSTFT) Loss** [15]: A measure of spectral similarity, where lower values (↓) are better.
- **Delta Metrics (△SI-SDR↑, △MRSTFT↓)**: To quantify the improvement, the change in SI-SDR and MRSTFT is reported, calculated as the final metric value

minus the initial metric value of the distorted (wet) test data before processing. Positive ΔSI-SDR and negative ΔMRSTFT indicate improvement.

A key aspect of the analysis involved cross-configuration comparisons, where models trained under one set of conditions were evaluated on test sets generated with different conditions to assess generalization and robustness. Table 5 summarizes the key characteristics of the datasets generated by the different configurations used for training and testing, including the total duration of the underlying clean audio and the initial quality metrics of the test sets after augmentation and normalization. Appendix 2 provides a detailed tables of the results obtained from the experiments, including the performance metrics for each configuration across different test sets.

Table 5. Experiments and Datasets: Characteristics and Quality Metrics

Configuration (augmentation, normalization)	**Duration**	**MRSTFT ↓**	**SI-SDR ↑ (dB)**
Electric (Tanh, Peak)	10h 17m	8.62	1.89
Electric (Tanh, Loudness -23 LUFS)	10h 17m	2.24	3.13
Electric (Tanh, Loudness -32 LUFS)	10h 17m	1.23	7.52
Acoustic (Tanh, Loudness -32 LUFS)	4h 9m	1.02	7.78
Combined (Tanh, Loudness -32 LUFS)	14h 26m	1.17	7.58
Monophonic (Tanh, Loudness -32 LUFS)	2h 57m	1.18	8.29
Polyphonic (Tanh, Loudness -32 LUFS)	11h 29m	1.14	7.59
GuitarSet (Tanh, Loudness -32 LUFS)	3h 3m	0.99	8.05
EGDB (Tanh, Loudness -32 LUFS)	1h 47m	0.96	8.92
Guitar-TECHS (Tanh, Loudness -32 LUFS)	5h 1m	1.28	8.11
IDMT (Tanh, Loudness -32 LUFS)	4h 34m	1.22	7.00
Acoustic (NN, Loudness -32 LUFS)	4h 9m	1.58	-10.05
Electric (NN, Loudness -32 LUFS)	10h 17m	1.68	-13.80
Combined (NN, Loudness -32 LUFS)	14h 26m	1.57	-14.23

4.1 Normalization Strategy: Peak and Loudness

This experiment investigates the impact of peak versus perceptual loudness normalization on model performance. Peak normalization scales the signal to a fixed maximum amplitude (e.g., -1.0 dBFS, ensuring values are within [-1, 1]), while loudness normalization (using `pyloudnorm` [7]) targets a specific perceived loudness (e.g., -23 or -32 LUFS). The latter may result in peak values exceeding the [-1, 1] range, especially after applying effects

like distortion, which can be problematic for numerical stability in deep learning models. It was observed that loudness normalization targets leading to clipped samples (like -23 LUFS in this case, as seen by its initial SI-SDR of 3.13 dB in Table 5 compared to 7.52 dB for -32 LUFS) might hinder training compared to targets ensuring no clipping (-32 LUFS) or peak normalization. Models were trained on electric guitar data (`tanh` augmentation) with each normalization strategy and evaluated across all strategies.

The results, detailed in Tables 6 and 7 and illustrated in Figures 4 and 5, confirm the importance of matching normalization strategies between training and testing phases and support the hypothesis regarding numerical stability. The loudness normalization to -32 LUFS, which avoided clipping in the initial data (initial SI-SDR 7.52 dB, MRSTFT 1.23 from Table 5), achieved the best performance when training and testing conditions matched. This configuration yielded the highest SI-SDR (**25.84 dB**) and the lowest MRSTFT loss (**0.22**). As seen in Figure 4, this also corresponded to the largest improvement over the initial distorted signal, with a ΔSI-SDR of **+18.31 dB** and a ΔMRSTFT of **-1.01** (Figure 5).

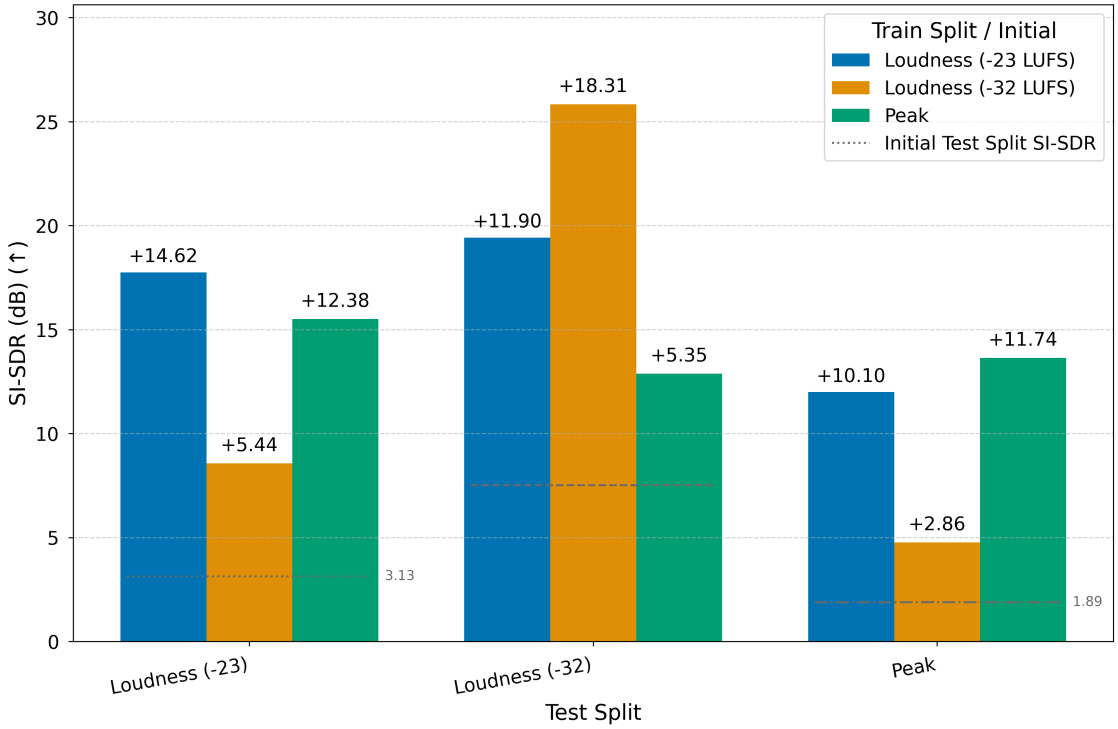

Figure 4. Performance by Normalization Strategy: SI-SDR / ΔSI-SDR (dB) ↑

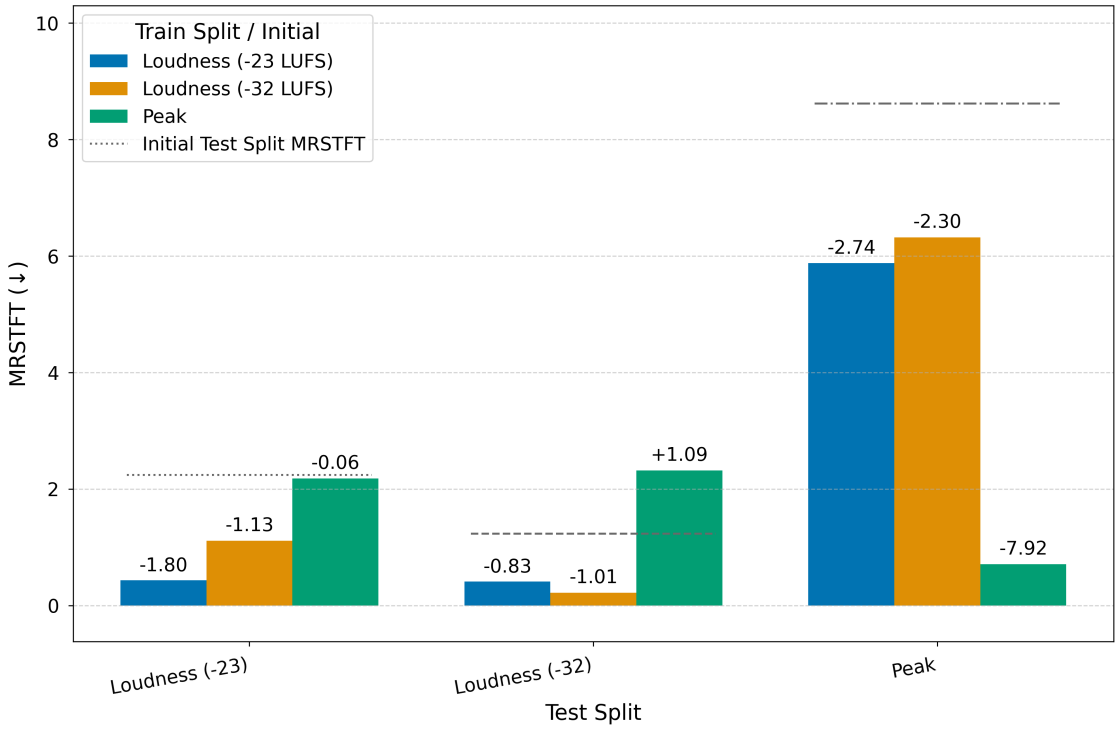

Figure 5. Performance by Normalization Strategy: MRSTFT / ΔMRSTFT ↓

The -23 LUFS target, which resulted in clipped initial samples (initial SI-SDR 3.13 dB, MRSTFT 2.24), led to significantly lower matched performance (17.75 dB SI-SDR, 0.43 MRSTFT) and smaller improvements (ΔSI-SDR +14.62 dB, ΔMRSTFT -1.80). This is likely due to numerical instability caused by clipping during training or testing. Peak normalization, with an initial SI-SDR of 1.89 dB and a very high initial MRSTFT of 8.62, yielded the lowest matched SI-SDR (13.63 dB) among the three strategies. Although its MRSTFT score (0.71) showed a large improvement (ΔMRSTFT of -7.92, visible in Figure 5), this substantial spectral improvement did not translate to high signal fidelity, as indicated by the low SI-SDR.

Cross-condition testing, where the normalization strategy differed between training and testing, consistently showed significant performance degradation across both metrics. For instance, training with -32 LUFS normalization but testing on peak-normalized data resulted in a very low SI-SDR of 4.76 dB and a high MRSTFT of 6.32. This is clearly visualized in Figures 4 and 5 by the off-diagonal elements (e.g., Loudness (-32 LUFS) train split tested on Peak test split). Listening tests aligned with these objective results: models trained on -32 LUFS loudness normalized data effectively removed distortion, while models trained on peak-normalized data introduced audible artifacts, and models trained on -23 LUFS data showed less effective distortion removal. Therefore, -32 LUFS loudness normalization was adopted for subsequent experiments due to its superior performance

and stability under matched conditions.

4.2 Data Composition: Acoustic, Electric, and Combined

This experiment investigates how the type of guitar recording used for training affects performance. Models trained exclusively on acoustic data (GuitarSet, IDMT-SMT-Guitar acoustic data), solely on electric data (EGDB, IDMT-SMT-Guitar electric data, GuitarTECHS), or on a combination of both are compared. All configurations use -32 LUFS loudness normalization and `tanh` augmentation. The initial characteristics of these test sets (Acoustic, Electric, Combined with Tanh, Loudness -32 LUFS) can be found in Table 5.

Tables 8 and 9, along with Figures 6 and 7, demonstrate that training on the Combined dataset (acoustic and electric guitars) yields the best overall generalization. While the model trained exclusively on acoustic data performs best when tested on matched acoustic data (SI-SDR **24.06 dB**, MRSTFT **0.20**, ΔSI-SDR **+16.28 dB**, ΔMRSTFT **-0.82**), its performance drops significantly when tested on electric data (SI-SDR 21.40 dB) or combined data (SI-SDR 22.03 dB). This is evident in Figure 6, where the model trained on Acoustic data has highest performance for the Acoustic test split but lower for Electric and Combined test splits.

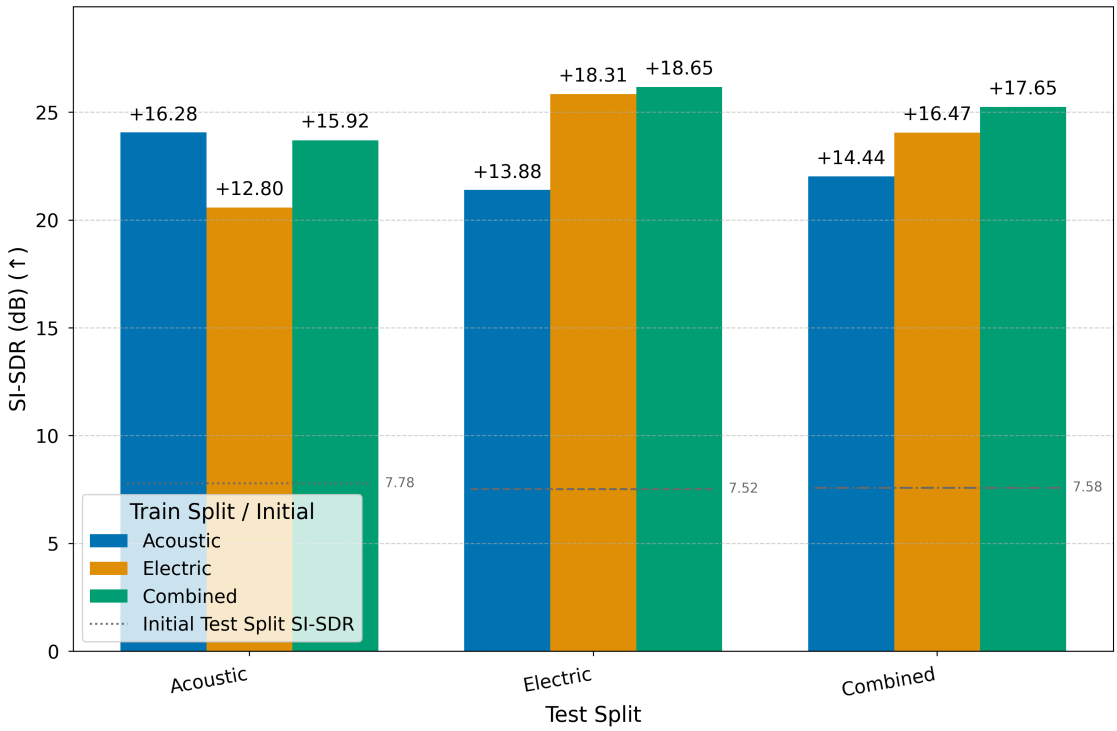

Figure 6. Performance by Data Composition (Acoustic, Electric, Combined): SI-SDR / ΔSI-SDR (dB) ↑

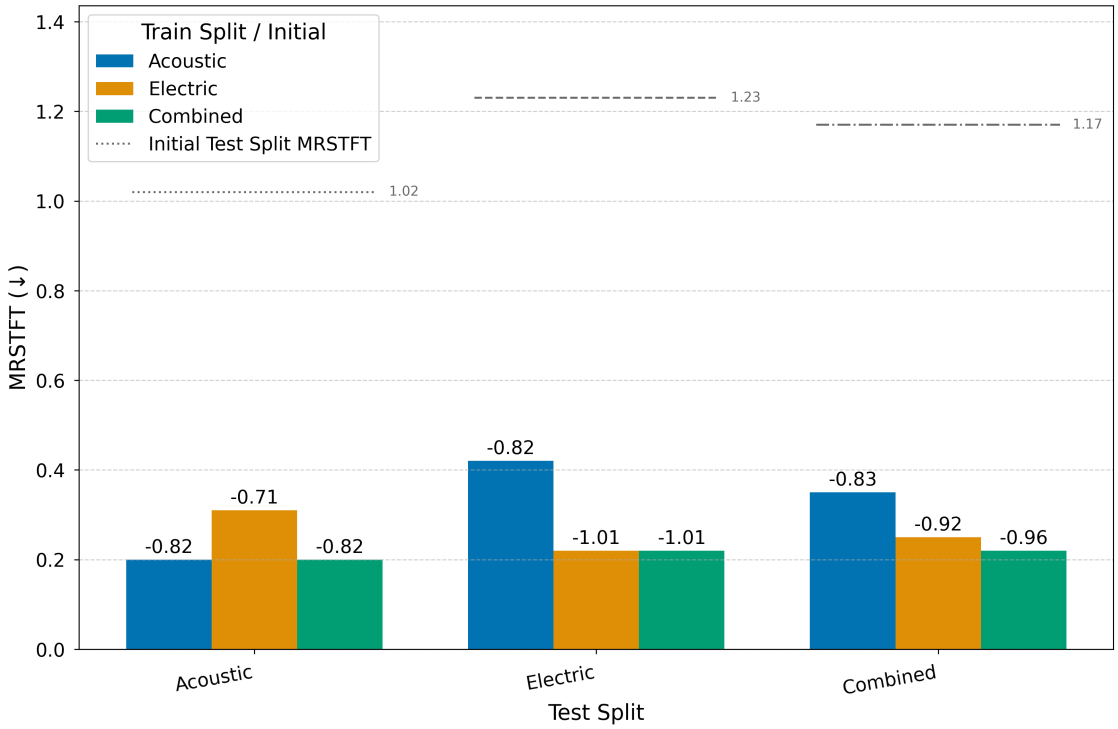

Figure 7. Performance by Data Composition (Acoustic, Electric, Combined): MRSTFT / ΔMRSTFT ↓

Conversely, the model trained on the Combined dataset achieves the highest SI-SDR and lowest MRSTFT scores, along with the largest improvements, when tested on electric data (SI-SDR **26.17 dB**, MRSTFT **0.22**, ΔSI-SDR **+18.65 dB**, ΔMRSTFT **-1.01**) and combined data (SI-SDR **25.24 dB**, MRSTFT **0.22**, ΔSI-SDR **+17.65 dB**, ΔMRSTFT **-0.96**). Furthermore, the Combined model maintains strong performance on purely acoustic test data (SI-SDR 23.70 dB, MRSTFT 0.20), performing nearly as well as the specialized Acoustic model. Figure 6 clearly shows that model trained on Combined data achieved top or near-top performance across all three test splits. This indicates that exposing the model to both clean acoustic and diverse electric guitar sounds during training is crucial for building a more robust and generalizable effect removal system capable of handling varied inputs.

4.3 Data Musical Texture: Monophonic, Polyphonic, and Combined Data

This experiment evaluates performance based on musical texture, comparing models trained on Monophonic, Polyphonic, or Combined datasets (-32 LUFS loudness normalization, `tanh` augmentation). The initial characteristics of the corresponding test sets (Monophonic, Polyphonic, Combined with Tanh, Loudness -32 LUFS) can be found in

Table 5.

Similar to the findings regarding acoustic versus electric data, the results presented in Tables 10 and 11 and visualized in Figures 8 and 9 show that training on combined data (both monophonic and polyphonic textures) leads to the best overall generalization. As seen in Figure 8, the Combined model achieves the highest SI-SDR and largest improvement on monophonic test data (SI-SDR **26.11 dB**, ΔSI-SDR **+17.81 dB**) and combined test data (SI-SDR **25.24 dB**, MRSTFT **0.22**, ΔSI-SDR **+17.65 dB**, ΔMRSTFT **-0.96**, see also Figure 9).

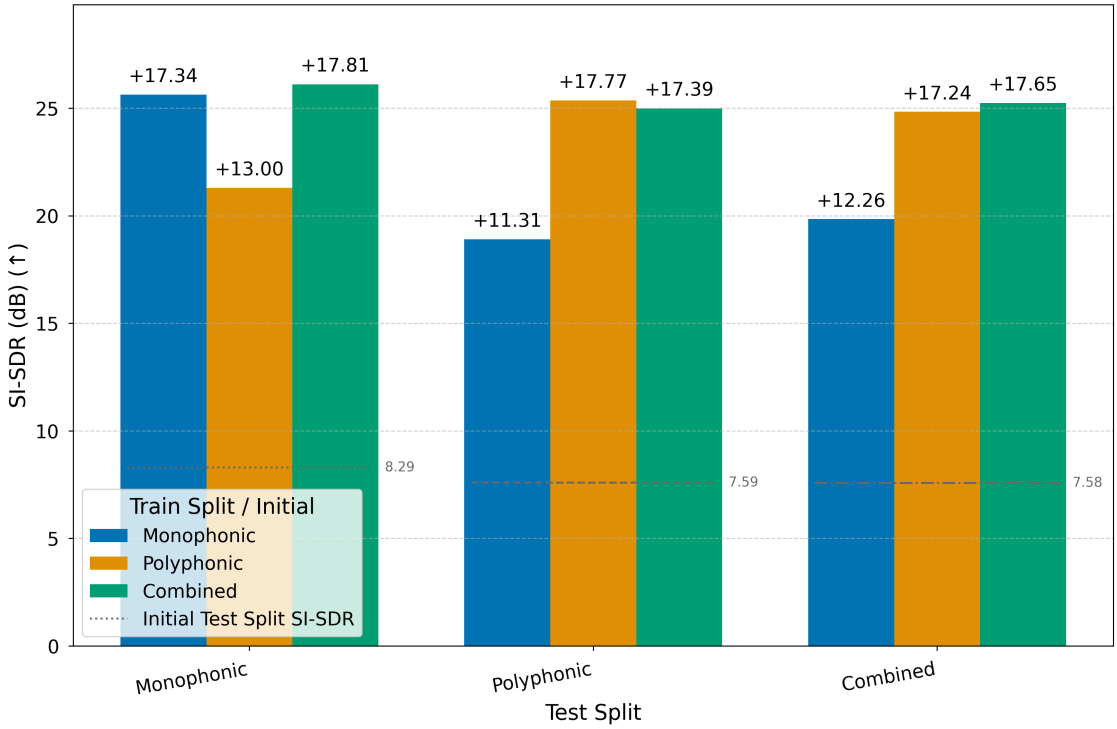

Figure 8. Performance by Data Musical Texture (Monophonic, Polyphonic, Combined): SI-SDR / ΔSI-SDR (dB) ↑

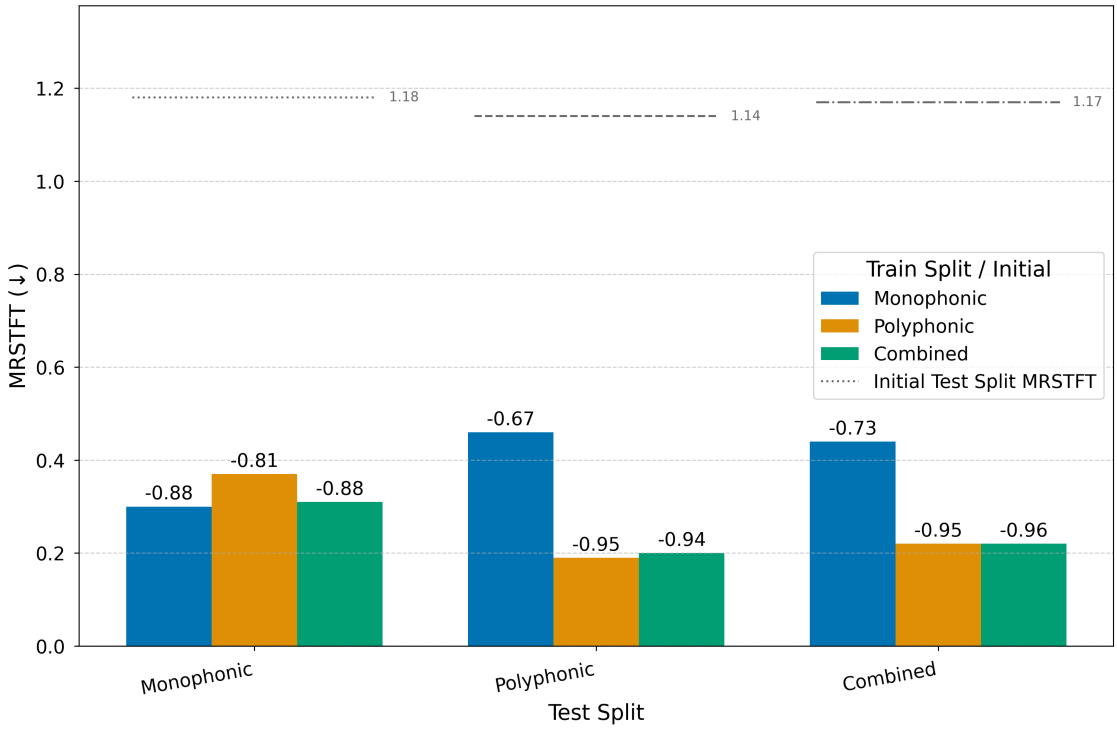

Figure 9. Performance by Data Musical Texture (Monophonic, Polyphonic, Combined): MRSTFT / ΔMRSTFT ↓

While the model trained only on polyphonic data performs best on the matched Polyphonic test set (SI-SDR **25.36 dB**, MRSTFT **0.19**, ΔSI-SDR **+17.77 dB**, ΔMRSTFT **-0.95**), the Combined model performs nearly as well (SI-SDR 24.99 dB, MRSTFT **0.20**, ΔSI-SDR +17.39 dB, ΔMRSTFT **-0.94**). Models trained exclusively on either monophonic or polyphonic data exhibit significant performance degradation when tested on the other texture type (e.g., for the Monophonic model on the Polyphonic test split in Figure 8, resulting in only 18.90 dB SI-SDR). This underscores the importance of including diverse musical textures during training to enhance model robustness and generalization across different input types.

4.4 Data Aggregation: Individual and Combined Datasets

This experiment directly compares models trained on individual datasets versus aggregated datasets (-32 LUFS loudness normalization, `tanh` augmentation), highlighting the benefits of data aggregation. The five training configurations evaluated are: μ Acoustic, representing the average performance of models trained separately on GuitarSet and IDMT-SMT-Guitar acoustic data; Acoustic, a model trained on combined acoustic data; μ Electric, the average performance of models trained separately on EGDB, Guitar-TECHS, and IDMT-SMT-Guitar electric data; Electric, a model trained on combined electric data; and Combined,

a model trained on all available datasets. These models are tested against corresponding data splits, allowing a detailed view of generalization capabilities. μ Acoustic averages results on GuitarSet and IDMT-SMT-Guitar (acoustic) test sets, while μ Electric averages results on EGDB, Guitar-TECHS, and IDMT-SMT-Guitar (electric) test sets.

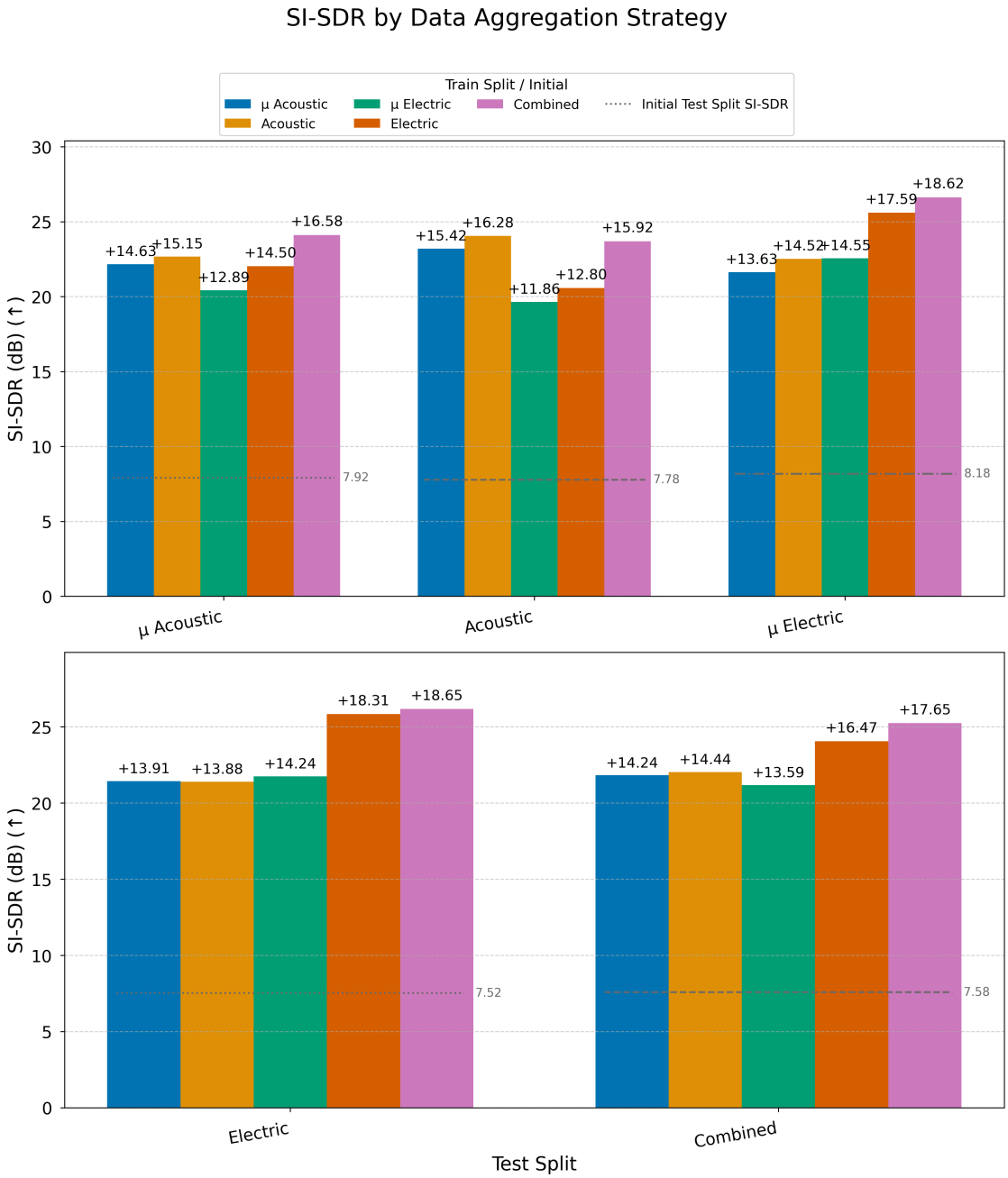

Figure 10. Performance by Data Aggregation Strategy (Individual vs. Combined): SI-SDR / ΔSI-SDR (dB) ↑

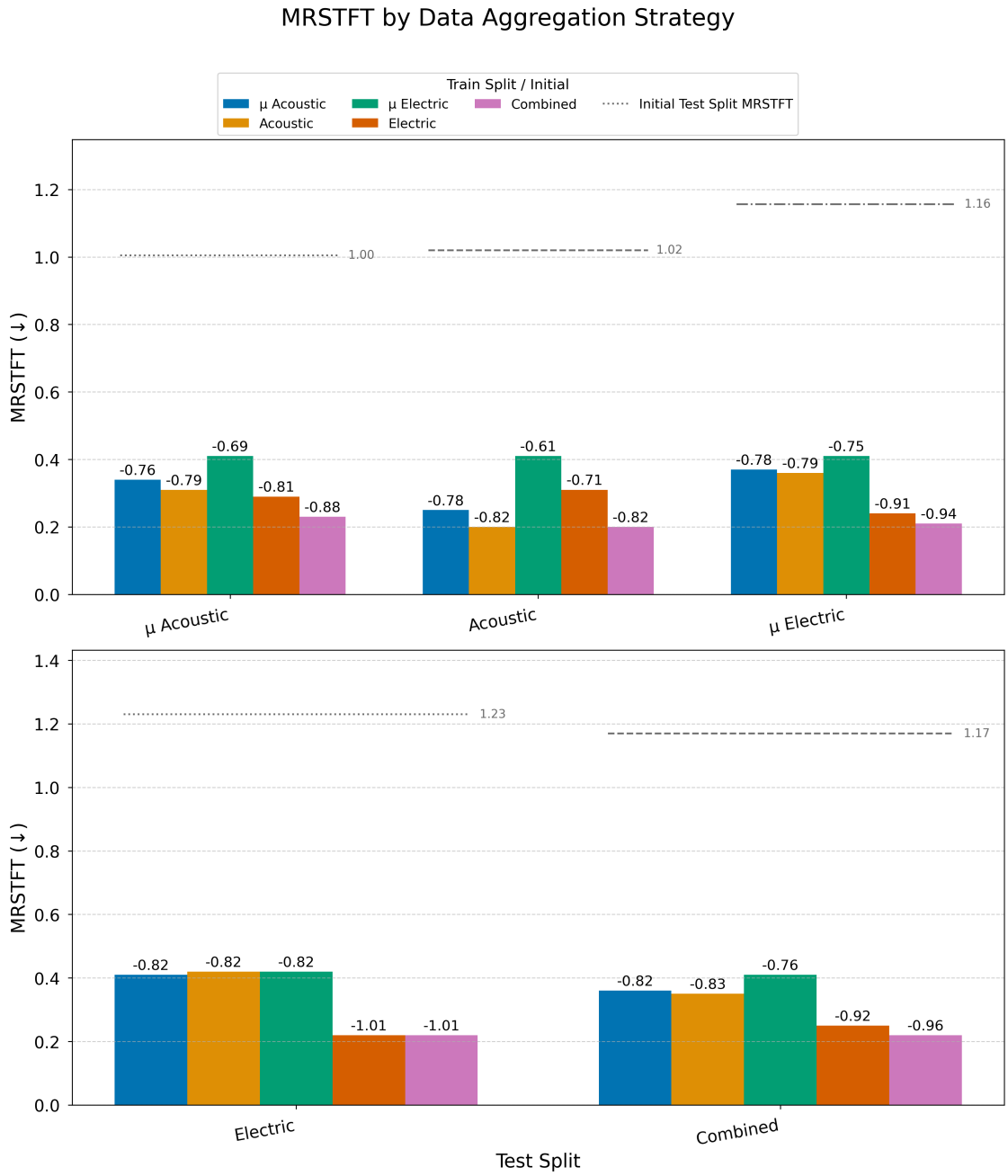

Figure 11. Performance by Data Aggregation Strategy (Individual vs. Combined): MRSTFT / ΔMRSTFT ↓

The results, detailed in Tables 12 and 13 and visualized in Figures 10 and 11, strongly reinforce the benefits of aggregating diverse datasets. The Combined model, trained on all available data, consistently demonstrates superior performance and generalization compared to models trained on individual datasets or averaged results from individually trained models.

Specifically, the Combined model achieves the highest SI-SDR scores and lowest MRSTFT

scores (indicating best performance) across multiple aggregated test conditions:

- μ **Acoustic Test:** SI-SDR **24.11 dB** (Δ **+16.58 dB**), MRSTFT **0.23** (Δ **-0.88**)
- μ **Electric Test:** SI-SDR **26.63 dB** (Δ **+18.62 dB**), MRSTFT **0.21** (Δ **-0.94**)
- **Electric Test:** SI-SDR **26.17 dB** (Δ **+18.65 dB**), MRSTFT **0.22** (Δ **-1.01**)
- **Combined Test:** SI-SDR **25.24 dB** (Δ **+17.65 dB**), MRSTFT **0.22** (Δ **-0.96**)

While models trained on specific subsets (e.g., the Acoustic model) perform best on perfectly matched test data (Acoustic test: SI-SDR 24.06 dB, MRSTFT 0.20), their performance significantly degrades when evaluated on out-of-domain data (e.g., Acoustic model on Electric test data: SI-SDR 21.40 dB, in Figure 10, Part 2). Averaging the performance of models trained on individual datasets (μ Acoustic, μ Electric) generally results in lower performance than training a single model on the combined data for that category (Acoustic, Electric). The Combined model's consistent high performance across diverse test sets, clearly visible in Figures 10 and 11, highlights the significant advantage of training on a large, aggregated dataset encompassing varied acoustic and electric guitar characteristics from multiple sources.

4.5 Augmentation Strategy: Baseline and Advanced

This experiment compares the baseline `tanh` augmentation against the more advanced neural network (NN) based augmentation using OpenAmp models [10]. The results across acoustic, electric, and combined training data for each augmentation type are averaged (μ Advanced (NN), μ Baseline (Tanh)) and evaluated on test sets created with both augmentation methods (-32 LUFS loudness normalization). The initial quality metrics for the NN-augmented and Tanh-augmented test sets can be found in Table 5.

Tables 14 and 15, along with Figures 12 and 13, reveal a significant domain mismatch between the two augmentation techniques. Models perform well only when the training augmentation matches the testing augmentation. Baseline trained models achieve high performance on Baseline augmented test data (SI-SDR **24.53 dB**, MRSTFT **0.26**, ΔSI-SDR **+16.90 dB**, ΔMRSTFT **-0.89**, but fail completely when tested on Advanced augmented test data (SI-SDR -11.88 dB, MRSTFT 1.60, Figure 12 shows minimal ΔSI-SDR improvement of +0.81 dB, while Figure 13 shows MRSTFT improvement of -0.02).

Conversely, Advanced trained models perform best on Advanced augmented test data (SI-SDR **11.28 dB**, MRSTFT **0.75**) but perform poorly on Baseline augmented test data (SI-SDR -1.09 dB, MRSTFT 1.07). Notably, while the absolute SI-SDR achieved by Advanced trained models on matched Advanced data (11.28 dB) is lower than that of

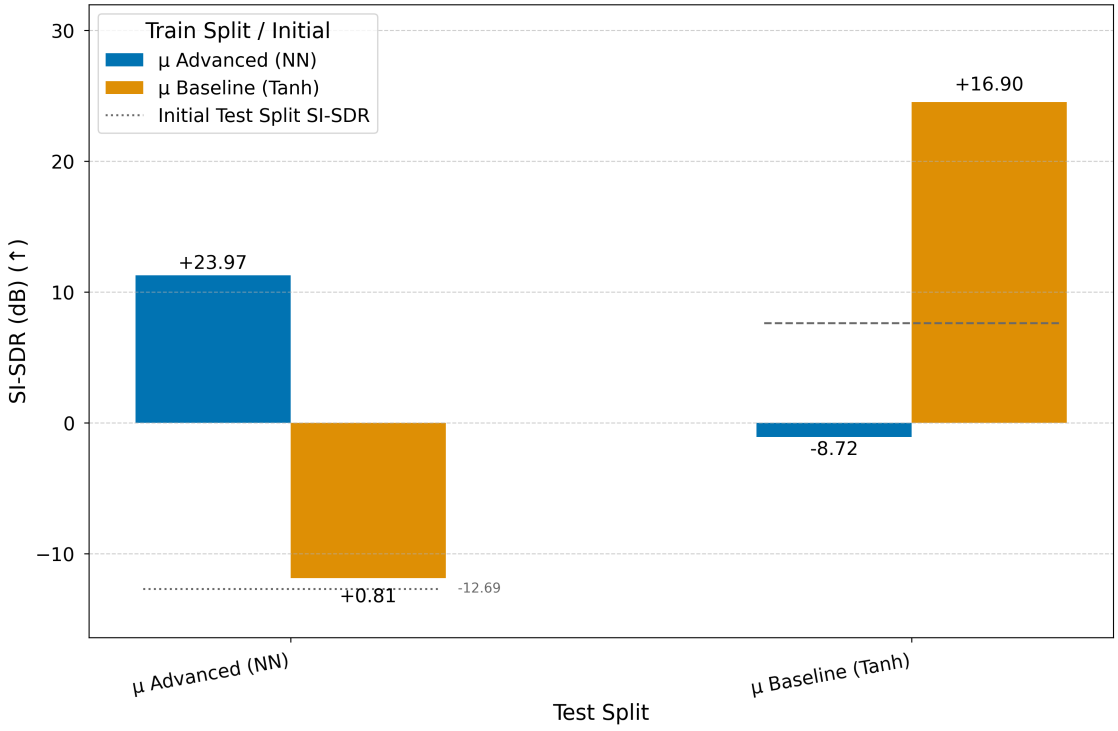

Figure 12. Performance by Augmentation Strategy (Baseline vs. Advanced): SI-SDR / ΔSI-SDR (dB) ↑

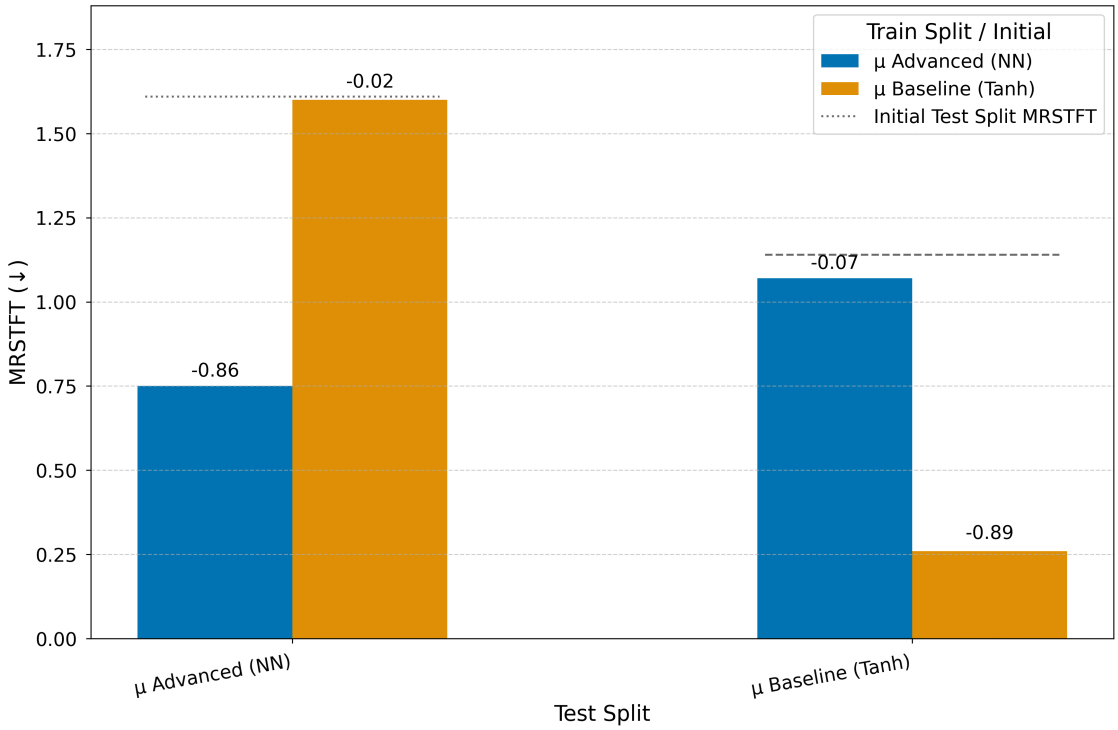

Figure 13. Performance by Augmentation Strategy (Baseline vs. Advanced): MRSTFT / ΔMRSTFT ↓

Baseline trained models on matched Baseline data (24.53 dB), the improvement (ΔSI-SDR) is substantially larger for the Advanced case (**+23.97 dB**) compared to the Baseline case (+16.90 dB), as clearly visualized in Figure 12. This suggests that the Advanced augmentation presents a significantly more challenging effect removal task (reflected in the lower absolute scores and worse initial SI-SDR values around -13 dB, see Table 5), but the model trained on this data learned to reverse these complex distortions more effectively, achieving a greater relative improvement.

These results strongly indicate that the choice of augmentation strategy is critical and must closely align with the characteristics of the target effects for successful removal. Simple augmentations like `tanh` may not adequately prepare models for the complexity of real-world distortions, which are potentially better simulated by the more advanced NN-based augmentation, even if it leads to lower absolute performance metrics in current models due to the increased task difficulty.

5. Results

This chapter presents the key findings derived from the series of experiments detailed in previous chapter. The experiments systematically evaluated the impact of different data aggregation, normalization, and augmentation strategies facilitated by the developed toolkit on the performance of a guitar effect removal model (RemFx Hybrid Demucs), using the SI-SDR and MRSTFT metrics. The results consistently highlight the benefits of specific preprocessing choices and diverse data aggregation for enhancing model robustness and generalization, as visualized in Figures 4 through 13 and detailed in Tables 6 through 15 (Appendix 2).

5.1 Impact of Normalization Strategy

The comparison between peak normalization and perceptual loudness normalization (at -23 LUFS and -32 LUFS targets), detailed in Section 4.1, revealed critical insights into maintaining numerical stability and perceptual consistency. As shown in Tables 6 and 7 (Appendix 2) and Figures 4 and 5, the loudness normalization strategy targeting -32 LUFS achieved superior performance when the training and testing normalization methods were matched. This configuration yielded the highest SI-SDR (25.84 dB) and the lowest MRSTFT loss (0.22), corresponding to the largest performance improvement over the initial distorted audio (ΔSI-SDR $+18.31$ dB, ΔMRSTFT -1.01).

Conversely, the -23 LUFS target, which led to clipping in the input data (Table 5), resulted in significantly lower matched performance (SI-SDR 17.75 dB), likely due to numerical instability. Peak normalization, while showing a substantial spectral improvement (ΔMRSTFT -7.92), yielded the lowest matched SI-SDR (13.63 dB), indicating poor signal fidelity despite spectral correction.

Crucially, mismatching normalization methods between training and testing led to severe performance degradation across all metrics, as visualized by the off-diagonal results in Figures 4 and 5. For instance, training with -32 LUFS normalization but testing on peak-normalized data resulted in an SI-SDR of only 4.76 dB. These quantitative results, corroborated by listening tests mentioned in Section 4.1, underscore the importance of selecting a normalization strategy that avoids clipping (like -32 LUFS loudness normalization) and maintaining consistency between training and evaluation phases. Consequently, -32 LUFS loudness normalization was adopted for all subsequent experiments.

5.2 Influence of Data Composition: Acoustic vs. Electric

Investigating the role of data source diversity (acoustic vs. electric guitar recordings), as described in Section 4.2, demonstrated the clear advantage of training on a combined dataset. Tables 8 and 9 (Appendix 2), along with Figures 6 and 7, show that while models trained exclusively on one data type (e.g., acoustic) performed best on matched test data (Acoustic test: SI-SDR 24.06 dB), their performance dropped considerably when evaluated on the other data type (Acoustic model on Electric test: SI-SDR 21.40 dB).

The model trained on the combined acoustic and electric data achieved the best overall generalization. It attained the highest SI-SDR scores when tested on electric data (26.17 dB) and combined data (25.24 dB), along with the largest performance improvements (ΔSI-SDR $+18.65$ dB and $+17.65$ dB, respectively). Furthermore, as seen in Figure 6, this combined model maintained strong performance on purely acoustic test data (SI-SDR 23.70 dB), performing almost as well as the specialized acoustic model. This highlights that exposure to diverse timbres from both acoustic and electric guitars during training is essential for building a robust effect removal system capable of handling varied inputs.

5.3 Influence of Musical Texture: Monophonic vs. Polyphonic

Similar conclusions were drawn from evaluating the impact of musical texture (Section 4.3). As presented in Tables 10 and 11 (Appendix 2) and visualized in Figures 8 and 9, training on a dataset combining both monophonic and polyphonic examples resulted in the best overall generalization. The combined model achieved the highest SI-SDR on monophonic test data (26.11 dB, ΔSI-SDR $+17.81$ dB) and combined test data (25.24 dB, ΔSI-SDR $+17.65$ dB).

While the model trained exclusively on polyphonic data performed best on the matched polyphonic test set (SI-SDR 25.36 dB), the combined model performed nearly as well (SI-SDR 24.99 dB). Models trained solely on one texture type showed significant performance degradation when tested on the other (e.g., the monophonic-trained model achieved only 18.90 dB SI-SDR on polyphonic data, visible in Figure 8). This emphasizes the necessity of including both monophonic and polyphonic examples during training to ensure the model can effectively process different musical textures encountered in real-world audio.

5.4 Benefit of Dataset Aggregation

The direct comparison between models trained on individual datasets versus aggregated datasets (Section 4.4) strongly validated the aggregation approach facilitated by the toolkit. Tables 12 and 13 (Appendix 2), along with Figures 10 and 11, clearly show that the Combined model, trained on all available datasets (IDMT-SMT-Guitar, GuitarSet, EGDB, Guitar-TECHS), consistently outperformed models trained on individual datasets or even models trained on combined subsets (like Acoustic or Electric).

The Combined model achieved the highest SI-SDR and lowest MRSTFT scores across nearly all aggregated test conditions, including average acoustic (24.11 dB SI-SDR), average electric (26.63 dB SI-SDR), electric (26.17 dB SI-SDR), and combined (25.24 dB SI-SDR). While specialized models performed best only on perfectly matched data, their generalization was poor. Averaging the results of individually trained models (μ Acoustic, μ Electric) also yielded lower performance than training a single model on the aggregated data. This demonstrates the significant advantage of leveraging the toolkit's aggregation capabilities to train on a large, diverse dataset encompassing multiple sources, leading to superior robustness and generalization.

5.5 Comparison of Augmentation Techniques

The final set of experiments (Section 4.5) compared the baseline `tanh` augmentation with the advanced neural network (NN) based augmentation using OpenAmp models. The results, summarized in Tables 14 and 15 (Appendix 2) and visualized in Figures 12 and 13, indicated a substantial domain mismatch between the two methods. Models performed well only when the augmentation method used during training matched the one used for testing.

Models trained with `tanh` augmentation achieved a high absolute SI-SDR (24.53 dB) on `tanh`-augmented test data but failed completely when tested on NN-augmented data (SI-SDR -11.88 dB). Conversely, models trained with NN augmentation performed best on NN-augmented test data (SI-SDR 11.28 dB) but poorly on `tanh`-augmented data (SI-SDR -1.09 dB).

Interestingly, although the absolute SI-SDR was lower for the NN-trained model on matched data, the *improvement* (ΔSI-SDR) was significantly larger (+23.97 dB) compared to the `tanh`-trained model on matched data ($+16.90$ dB), as highlighted in Figure 12. This suggests that the NN-based augmentation, while representing a more challenging task

(reflected in lower absolute scores and worse initial SI-SDR values in Table 5), enables the model to learn to reverse more complex, realistic distortions more effectively. The choice of augmentation is therefore critical; simple methods like `tanh` may not adequately prepare models for real-world effect removal, whereas advanced NN-based methods, despite potentially lowering absolute scores with current model architectures due to increased task difficulty, offer a path towards handling more complex distortions.

In summary, the presented experimental results validate the design choices implemented in the toolkit. Utilizing -32 LUFS loudness normalization, aggregating data from diverse sources (acoustic/electric, mono/poly, multiple datasets), and employing augmentation techniques that reflect the complexity of real-world effects (like the NN-based approach) are crucial steps towards building robust and generalizable guitar audio effect removal systems. The toolkit effectively facilitates these steps, providing a strong foundation for future research in this domain.

6. Conclusion

This thesis addressed the critical need for robust data aggregation and generation methodologies in the domain of guitar audio processing, particularly focusing on the challenging task of audio effect removal. Recognizing the limitations imposed by the scarcity of diverse, large-scale datasets and standardized preprocessing pipelines, a comprehensive toolkit designed to bridge this gap was developed. The toolkit facilitates the aggregation of heterogeneous guitar datasets (IDMT-SMT-Guitar, GuitarSet, EGDB, Guitar-TECHS), implements efficient preprocessing techniques, and offers flexible, on-the-fly normalization and augmentation strategies. Leveraging modern frameworks like PyTorch, Hydra, and PyTorch Lightning, the toolkit provides a significant improvement over existing solutions, offering features specifically tailored for effect removal research.

The research presented herein aimed to answer key questions regarding data preparation for guitar effect removal models. A series of systematic experiments, detailed in Chapter 4 using the RemFx framework, provided the following insights:

1. Effective Methods for Increasing Data Size and Diversity: The experiments demonstrated conclusively that aggregating all available diverse datasets (IDMT-SMT-Guitar, GuitarSet, EGDB, Guitar-TECHS) into a single, large training pool is the most effective method identified in this study for increasing size and diversity (Section 4.4). The model trained on this fully aggregated "Combined" dataset consistently outperformed models trained on individual datasets or smaller aggregated subsets (e.g., only acoustic or only electric) across nearly all test conditions, achieving superior generalization and the highest average performance metrics (e.g., 26.63 dB SI-SDR on average electric test data, Tables 12, 13). This highlights the significant advantage of maximizing dataset diversity through aggregation.

2. Optimal Normalization Strategies: Regarding normalization (Section 4.1), perceptual loudness normalization targeting -32 LUFS proved to be the most effective strategy. This approach significantly outperformed peak normalization and loudness targets prone to clipping (-23 LUFS in the tests), yielding the highest SI-SDR (25.84 dB) and lowest MRSTFT (0.22) on matched test data (Tables 6, 7). The -32 LUFS target avoided clipping observed with louder targets, ensuring numerical stability. Crucially, maintaining consistency in the normalization method between training and testing phases was found to be essential for optimal performance, as mismatches led to drastic degradation.

3. Leveraging Augmentation Techniques: The study (Section 4.5) revealed a significant domain mismatch between baseline (`tanh`) and advanced NN-based (OpenAmp) augmentation. While `tanh`-trained models achieved higher absolute SI-SDR on matched data (24.53 dB), advanced NN-trained models demonstrated substantially larger relative improvements (ΔSI-SDR $+23.97$ dB vs. $+16.90$ dB) when tested on matched NN-augmented data (Tables 14, 15). This indicates that while NN augmentation presents a more difficult task (reflected in lower absolute scores currently), it likely simulates real-world effect complexity more realistically. Therefore, leveraging advanced NN-based augmentation is crucial for enhancing model generalization to complex, real-world distortions, even if it requires more capable model architectures to fully realize its potential.

4. Role of Data Categorization: The categorization of guitar audio data plays a critical role in the performance of distortion removal systems. Experiments focusing on source type (acoustic vs. electric, Section 4.2) and musical texture (monophonic vs. polyphonic, Section 4.3) both showed that training on maximally diverse data encompassing both acoustic and electric guitars, and both monophonic and polyphonic textures, yielded the most robust and generalizable models. Models trained on only one category (e.g., only acoustic, only polyphonic) suffered significant performance drops when tested on other categories (Tables 8, 9, 10, 11). This emphasizes that comprehensive categorization and inclusion of diverse data types during training are vital for robust performance.

In synthesis, the experimental results underscore the critical importance of a holistic data preparation strategy: utilizing perceptual loudness normalization applied consistently (-32 LUFS), maximizing data diversity through the aggregation of varied sources, textures, and datasets, and employing realistic augmentation techniques that capture the complexity of real-world effects.

The developed toolkit successfully implements these findings, providing a robust, reproducible, and extensible platform for guitar audio research. It effectively addresses the limitations of data scarcity and inconsistent processing, offering a practical tool and validated methodologies. This work bridges the gap between foundational MIR tasks and the less explored, yet crucial, area of effect removal, paving the way for future advancements in building more effective and generalizable guitar audio processing models.

7. Future Work

Building upon the foundation laid by this thesis, several promising avenues for future research and development emerge, primarily centered on enhancing the diversity and realism of the training data and broadening the scope of the toolkit and its applications. Firstly, the continued expansion of dataset support is crucial. Integrating additional publicly available guitar datasets into the toolkit's aggregation pipeline will further enrich data diversity, enabling more comprehensive model training and robust evaluation across an even wider range of recording conditions and playing styles. Alongside data expansion, ongoing toolkit enhancements, such as improved documentation, more illustrative examples, will increase its accessibility and utility for the wider research community.

A significant area for future development lies in refining the augmentation strategies. The current comparison between baseline `tanh` and advanced NN-based distortion highlighted the need for more nuanced and controllable approaches. Future work should focus on extending the NN-based augmentation module, or implementing alternatives, to allow for the explicit selection and weighting of different distortion characteristics (e.g., soft-clipping/overdrive, hard-clipping/distortion, fuzz). This aligns with findings in general effect removal [13] suggesting that effect-specific models can improve performance, and this concept can be extended within the distortion category itself. Furthermore, implementing mechanisms to control the intensity or grade of the applied distortion during augmentation would be highly beneficial. This would enable the training of models specialized not only for distortion type but also for distortion severity, potentially leading to significantly better performance across a wider spectrum of real-world scenarios. Exploring strategies for combining different augmentation methods, such as mixing `tanh` and NN-based effects or applying sequential effects, could also better simulate the complexity of real-world guitar pedal chains. The overarching goal of these refinements is to facilitate the training of more specialized effect removal models tailored to specific distortion profiles, which could potentially surpass the performance of general-purpose models on those particular tasks.

Finally, the research can be extended by exploring more advanced model architectures for the task of effect removal, evaluating how datasets generated by this toolkit perform with other state-of-the-art audio processing and source separation models beyond the Hybrid Demucs architecture used herein (e.g., diffusion models, newer Transformer variants). Moreover, the toolkit's augmentation capabilities and experimental validation should be

broadened to incorporate other common guitar effects such as reverb, delay, chorus, and compression. This would move towards a more comprehensive multi-effect removal framework, addressing a wider range of challenges in audio restoration and production. By pursuing these directions, the research community can further leverage and build upon the toolkit developed in this thesis to significantly advance the state-of-the-art in guitar audio effect removal and related Music Information Retrieval tasks.

References

[1] Qingyang Xi et al. “GuitarSet: A Dataset for Guitar Transcription”. In: *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*. Ed. by Emilia Gómez et al. 2018, pp. 453–460. URL: http://ismir2018.ircam.fr/doc/pdfs/188%5C_Paper.pdf (visited on 03/30/2025).

[2] Yu-Hua Chen et al. “Towards Automatic Transcription of Polyphonic Electric Guitar Music: A New Dataset and a Multi-Loss Transformer Model”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. May 2022, pp. 786–790. DOI: 10.1109/ICASSP43922.2022.9747697. URL: https://ieeexplore.ieee.org/document/9747697 (visited on 03/16/2025).

[3] Christian Kehling et al. “Automatic Tablature Transcription of Electric Guitar Recordings by Estimation of Score- and Instrument-Related Parameters”. In: *Proceedings of the 17th International Conference on Digital Audio Effects, DAFx-14, Erlangen, Germany, September 1-5, 2014*. 2014. URL: http://www.dafx14.fau.de/papers/dafx14%5C_christian%5C_kehling%5C_automatic%5C_tablature%5C_trans.pdf (visited on 03/30/2025).

[4] Yongyi Zang et al. “SynthTab: Leveraging Synthesized Data for Guitar Tablature Transcription”. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. Apr. 2024, pp. 1286–1290. DOI: 10.1109/ICASSP48485.2024.10447902. URL: https://ieeexplore.ieee.org/document/10447902 (visited on 03/16/2025).

[5] Hegel Pedroza et al. “Guitar-TECHS: An Electric Guitar Dataset Covering Techniques, Musical Excerpts, Chords and Scales Using a Diverse Array of Hardware”. In: *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. Apr. 2025, pp. 1–5. DOI: 10.1109/ICASSP49660.2025.10887996. URL: https://ieeexplore.ieee.org/document/10887996 (visited on 03/16/2025).

[6] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by Francis R. Bach and David M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 448–456. URL: http://proceedings.mlr.press/v37/ioffe15.html (visited on 03/30/2025).

[7] Christian J. Steinmetz and Joshua Reiss. “pyloudnorm: A simple yet flexible loudness meter in python”. In: *Audio Engineering Society Convention 150*. Audio Engineering Society, 2021. URL: https://www.aes.org/e-lib/browse.cfm?elib=21076 (visited on 03/16/2025).

[8] Spotify AB. *Pedalboard*. URL: https://spotify.github.io/pedalboard/ (visited on 03/16/2025).

[9] Johannes Imort et al. “Distortion Audio Effects: Learning How to Recover the Clean Signal”. In: *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*. Ed. by Preeti Rao et al. 2022, pp. 218–225. URL: https://archives.ismir.net/ismir2022/paper/000025.pdf (visited on 03/30/2025).

[10] Alec Wright, Alistair Carson, and Lauri Juvela. “Open-Amp: Synthetic Data Framework for Audio Effect Foundation Models”. In: *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. Apr. 2025, pp. 1–5. DOI: 10.1109/ICASSP49660.2025.10888232. URL: https://ieeexplore.ieee.org/document/10888232 (visited on 03/16/2025).

[11] K. Bloemer. *Guitar ML: Tone Library*. URL: https://guitarml.com/tonelibrary/tonelib-pro.html (visited on 03/16/2025).

[12] S. Atkinson. *Neural Amp Modeler*. URL: https://www.neuralampmodeler.com/ (visited on 03/16/2025).

[13] Matthew Rice et al. “General Purpose Audio Effect Removal”. In: *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). ISSN: 1947-1629. Oct. 2023, pp. 1–5. DOI: 10.1109/WASPAA58266.2023.10248157. URL: https://ieeexplore.ieee.org/document/10248157 (visited on 03/16/2025).

[14] Jonathan Le Roux et al. “SDR – Half-baked or Well Done?” In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. May 2019, pp. 626–630. DOI: 10.1109/ICASSP.2019.8683855. URL: https://ieeexplore.ieee.org/document/8683855 (visited on 04/20/2025).

[15] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. “Parallel Wavegan: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram”. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. May 2020, pp. 6199–6203. DOI: 10.1109/ICASSP40776.2020.9053795. URL: https://ieeexplore.ieee.org/document/9053795 (visited on 04/20/2025).

[16] Hegel Pedroza and Wallace Abreu. “Leveraging Real Electric Guitar Tones and Effects to Improve Robustness in Guitar Tablature Transcription Modeling”. In: *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24), Guildford, United Kingdom, 3 - 7 Sept. 2024 (LBR)*. 2024. URL: https://www.dafx.de/paper-archive/2024/papers/DAFx24_paper_99.pdf (visited on 03/30/2025).

[17] Osamu Take et al. “Audio Effect Chain Estimation and Dry Signal Recovery From Multi-Effect-Processed Musical Signals”. In: *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24), Guildford, United Kingdom, 3 - 7 September 2024*. 2024. URL: https://www.dafx.de/paper-archive/2024/papers/DAFx24_paper_53.pdf (visited on 03/30/2025).

[18] Reemt Hinrichs et al. “Blind extraction of guitar effects through blind system inversion and neural guitar effect modeling”. In: *EURASIP J. Audio Speech Music. Process.* 2024.1 (2024), p. 9. DOI: 10.1186/S13636-024-00330-0. URL: https://doi.org/10.1186/s13636-024-00330-0 (visited on 03/30/2025).

[19] Alec Wright, Eero-Pekka Damskägg, and Vesa Välimäki. “Real-Time Black-Box Modelling With Recurrent Neural Networks”. In: *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19), Birmingham, UK, September 2–6, 2019*. 2019. URL: https://www.dafx.de/paper-archive/2019/DAFx2019_paper_43.pdf (visited on 03/30/2025).

[20] Yen-Tung Yeh et al. “DDSP Guitar Amp: Interpretable Guitar Amplifier Modeling”. In: *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech*

and Signal Processing (ICASSP). ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. Apr. 2025, pp. 1–5. DOI: 10.1109/ICASSP49660.2025.10889306. URL: https://ieeexplore.ieee.org/document/10889306 (visited on 03/16/2025).

[21] Andrew McLeod. "No Data Required: Zero-Shot Domain Adaptation for Automatic Music Transcription". In: *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. Apr. 2025, pp. 1–5. DOI: 10.1109/ICASSP49660.2025.10890396. URL: https://ieeexplore.ieee.org/document/10890396 (visited on 03/16/2025).

[22] Marco Comunita and J. Reiss. "AFxResearch: a repository and website of audio effects research". In: *DMRN+ 19: Digital Music Research Network One-day Workshop 2024*. 2024. DOI: 10.5281/zenodo.13380393. URL: https://qmro.qmul.ac.uk/xmlui/bitstream/handle/123456789/99119/Comunita%20AFX-Research:%20an%20Extensive%20and%20Flexible%20Repository%20of%20Research%20about%20Audio%20Effects%202024%20Published.pdf?sequence=2 (visited on 03/30/2025).

[23] Rachel M. Bittner et al. "mirdata: Software for Reproducible Usage of Datasets". In: *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*. Ed. by Arthur Flexer et al. 2019, pp. 99–106. URL: http://archives.ismir.net/ismir2019/paper/000009.pdf (visited on 03/30/2025).

[24] Joshua D. Reiss and Andrew McPherson. *Audio Effects*. CRC Press, 2014. ISBN: 978-1-4665-6028-4. URL: https://learning.oreilly.com/library/view/audio-effects/9781466560284/ (visited on 03/16/2025).

[25] Udo Zölzer. *DAFX: Digital Audio Effects, Second Edition*. Wiley, 2011. ISBN: 978-0-470-66599-2. URL: https://learning.oreilly.com/library/view/dafx-digital-audio/9780470665992/ (visited on 03/16/2025).

[26] Thomas Wilmering et al. "A History of Audio Effects". In: *Applied Sciences* 10.3 (Jan. 2020). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 791. ISSN: 2076-3417. DOI: 10.3390/app10030791. URL: https://www.mdpi.com/2076-3417/10/3/791 (visited on 03/16/2025).

[27] *JUCE: An open-source cross-platform C++ application framework*. 2023. URL: https://github.com/juce-framework/JUCE.

[28] Xavier Riley, Drew Edwards, and Simon Dixon. “High Resolution Guitar Transcription Via Domain Adaptation”. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. Apr. 2024, pp. 1051–1055. DOI: 10.1109/ICASSP48485.2024.10446182. URL: https://ieeexplore.ieee.org/abstract/document/10446182 (visited on 03/18/2025).

[29] Benoit Lachaise and Laurent Daudet. “Inverting dynamics compression with minimal side information”. In: *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08), Espoo, Finland, September 1-4, 2008*. 2008. URL: https://www.dafx.de/paper-archive/2008/papers/dafx08_18.pdf (visited on 03/30/2025).

[30] Stanislaw Gorlow and Joshua D. Reiss. “Model-Based Inversion of Dynamic Range Compression”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.7 (July 2013). Conference Name: IEEE Transactions on Audio, Speech, and Language Processing, pp. 1434–1444. ISSN: 1558-7924. DOI: 10.1109/TASL.2013.2253099. URL: https://ieeexplore.ieee.org/abstract/document/6480792 (visited on 03/16/2025).

[31] Srdan Kitic, Nancy Bertin, and Rémi Gribonval. “Sparsity and Cosparsity for Audio Declipping: A Flexible Non-convex Approach”. In: *Latent Variable Analysis and Signal Separation - 12th International Conference, LVA/ICA 2015, Liberec, Czech Republic, August 25-28, 2015, Proceedings*. Ed. by Emmanuel Vincent et al. Vol. 9237. Lecture Notes in Computer Science. Springer, 2015, pp. 243–250. DOI: 10.1007/978-3-319-22482-4_28. URL: https://doi.org/10.1007/978-3-319-22482-4%5C_28 (visited on 03/30/2025).

[32] Alberto Bernardini et al. “Towards Inverse Virtual Analog Modeling”. In: *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19), Birmingham, UK, September 2–6, 2019*. 2019. URL: https://www.dafx.de/paper-archive/2019/DAFx2019_paper_8.pdf (visited on 03/30/2025).

[33] Wolfgang Mack and Emanuël A. P. Habets. “Declipping Speech Using Deep Filtering”. In: *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). ISSN: 1947-1629. Oct. 2019, pp. 200–204. DOI: 10.1109/WASPAA.2019.8937287. URL: https://ieeexplore.ieee.org/abstract/document/8937287 (visited on 03/16/2025).

[34] Henrik Jürgens, Reemt Hinrichs, and Jörn Ostermann. “Recognizing Guitar Effects and Their Parameter Settings”. In: *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx-20), Vienna, Austria, September 8–12, 2020*. 2020. URL: https://www.dafx.de/paper-archive/2020/proceedings/papers/DAFx2020_paper_2.pdf (visited on 03/30/2025).

[35] Marco Comunità, Dan Stowell, and Joshua D. Reiss. “Guitar Effects Recognition and Parameter Estimation with Convolutional Neural Networks”. In: *Journal of the Audio Engineering Society* 69.7 (Nov. 11, 2021), pp. 594–604. ISSN: 15494950. DOI: 10.17743/jaes.2021.0019. arXiv: 2012.03216[cs]. URL: http://arxiv.org/abs/2012.03216 (visited on 03/18/2025).

[36] Tomoro Tanaka et al. “APPLADE: Adjustable Plug-and-Play Audio Declipper Combining DNN with Sparse Optimization”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. May 2022, pp. 1011–1015. DOI: 10.1109/ICASSP43922.2022.9747089. URL: https://ieeexplore.ieee.org/document/9747089 (visited on 03/16/2025).

[37] Ying-Shuo Lee et al. “Distortion Recovery: A Two-Stage Method for Guitar Effect Removal”. In: *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24), Guildford, United Kingdom, 3 - 7 September 2024*. 2024. URL: https://www.dafx.de/paper-archive/2024/papers/DAFx24_paper_59.pdf (visited on 03/30/2025).

[38] Michal Švento et al. *Estimation and Restoration of Unknown Nonlinear Distortion using Diffusion*. Jan. 10, 2025. DOI: 10.48550/arXiv.2501.05959. arXiv: 2501.05959[eess]. URL: http://arxiv.org/abs/2501.05959 (visited on 03/16/2025).

[39] Jayeon Yi, Junghyun Koo, and Kyogu Lee. “DDD: A Perceptually Superior Low-Response-Time DNN-Based Declipper”. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. Apr. 2024, pp. 801–805. DOI: 10.1109/ICASSP48485.2024.10446235. URL: https://ieeexplore.ieee.org/document/10446235 (visited on 03/30/2025).

[40] Yu-Hua Chen et al. “Improving Unsupervised Clean-to-Rendered Guitar Tone Transformation Using GANs and Integrated Unaligned Clean Data”. In: *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24) Guildford, Surrey, UK, September 3-7, 2024*. 2024. URL: https://www.dafx.de/

paper-archive/2024/papers/DAFx24_paper_30.pdf (visited on 03/30/2025).

[41] Thomas Schmitz and Jean-Jacques Embrechts. "Introducing a dataset of guitar amplifier sounds for nonlinear emulation benchmarking". In: *AES E-Library* (2018). URL: https://orbi.uliege.be/handle/2268/228910 (visited on 03/29/2025).

[42] Hegel Pedroza, Gerardo Meza, and Iran R. Roman. "EGFxSet: Electric guitar tones processed through real effects of distortion, modulation, delay and reverb". In: *ISMIR Late Breaking Demo* (2022). URL: https://archives.ismir.net/ismir2022/latebreaking/000006.pdf (visited on 03/30/2025).

[43] J. Abeßer et al. "Automatic Detection of Audio Effects in Guitar and Bass Recordings". In: *Audio Engineering Society (AES Convention) 2010*. 2010. URL: https://publica.fraunhofer.de/entities/publication/0576c30c-b56d-4105-9bb9-25e50367ac71/details (visited on 03/27/2025).

[44] Jort F. Gemmeke et al. "Audio Set: An ontology and human-labeled dataset for audio events". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. Mar. 2017, pp. 776–780. DOI: 10.1109/ICASSP.2017.7952261. URL: https://ieeexplore.ieee.org/document/7952261 (visited on 03/30/2025).

[45] Marios Glytsos. "Music source separation on classical guitar duets". BSc Thesis. National Technical University of Athens, 2024. URL: https://dspace.lib.ntua.gr/xmlui/bitstream/handle/123456789/60093/MusicSourceSeparationOnClassicalGuitarDuetsMariosThesisNTUA.pdf?sequence=1 (visited on 03/27/2025).

[46] Xavier Riley et al. "GAPS: A Large and Diverse Classical Guitar Dataset and Benchmark Transcription Model". In: *Proceedings of the 25th International Society for Music Information Retrieval Conference, ISMIR 2024, San Francisco, California, USA and Online, November 10-14, 2024*. Ed. by Blair Kaneshiro et al. 2024, pp. 611–617. DOI: 10.5281/ZENODO.14877413. URL: https://doi.org/10.5281/zenodo.14877413 (visited on 03/30/2025).

[47] Scott Hawley, Benjamin Colburn, and Stylianos Ioannis Mimilakis. "Profiling Audio Compressors with Deep Neural Networks". In: *Audio Engineering Society Convention 147*. Audio Engineering Society, 2019. URL: https://www.aes.org/e-lib/browse.cfm?elib=20595 (visited on 03/30/2025).

[48] Pedro Sarmento et al. “DadaGP: A Dataset of Tokenized GuitarPro Songs for Sequence Models”. In: *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*. Ed. by Jin Ha Lee et al. 2021, pp. 610–617. URL: https://archives.ismir.net/ismir2021/paper/000076.pdf (visited on 03/30/2025).

[49] Kumar Ashis Pati and Alexander Lerch. “A Dataset and Method for Guitar Solo Detection in Rock Music”. In: *AES International Conference Semantic Audio 2017, Erlangen, Germany, June 22-24, 2017*. Ed. by Christian Dittmar, Jakob Abeßer, and Meinard Müller. Audio Engineering Society, 2017. URL: http://www.aes.org/e-lib/browse.cfm?elib=18773 (visited on 03/30/2025).

[50] Alec Wright et al. “Real-Time Guitar Amplifier Emulation with Deep Learning”. In: *Applied Sciences* 10.3 (Jan. 2020). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 766. ISSN: 2076-3417. DOI: 10.3390/app10030766. URL: https://www.mdpi.com/2076-3417/10/3/766 (visited on 03/16/2025).

[51] Marco Comunità et al. “Modelling Black-Box Audio Effects with Time-Varying Feature Modulation”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. June 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10097173. URL: https://ieeexplore.ieee.org/document/10097173 (visited on 03/16/2025).

[52] Justin Salamon et al. “Scaper: A library for soundscape synthesis and augmentation”. In: *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). ISSN: 1947-1629. Oct. 2017, pp. 344–348. DOI: 10.1109/WASPAA.2017.8170052. URL: https://ieeexplore.ieee.org/document/8170052 (visited on 03/17/2025).

[53] Riccardo Simionato. “Fully Conditioned and Low-Latency Black-Box Modeling of Analog Compression”. In: *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx23), Copenhagen, Denmark, 4 - 7 September 2023*. 2022. URL: https://www.dafx.de/paper-archive/2023/DAFx23_paper_10.pdf (visited on 03/30/2025).

[54] David S. Johnson and Sascha Grollmisch. “Techniques Improving the Robustness of Deep Learning Models for Industrial Sound Analysis”. In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2020 28th European Signal Processing Conference (EUSIPCO). ISSN: 2076-1465. Jan. 2021, pp. 81–85. DOI: 10.23919/

Eusipco47968.2020.9287327. URL: https://ieeexplore.ieee.org/document/9287327 (visited on 04/01/2025).

[55] Yu-Hua Chen et al. "Towards Zero-Shot Amplifier Modeling: One-to-Many Amplifier Modeling via Tone Embedding Control". In: *Proceedings of the 25th International Society for Music Information Retrieval Conference, ISMIR 2024, San Francisco, California, USA and Online, November 10-14, 2024*. Ed. by Blair Kaneshiro et al. 2024, pp. 446–453. DOI: 10.5281/ZENODO.14877373. URL: https://doi.org/10.5281/zenodo.14877373 (visited on 03/30/2025).

[56] Yu Wang et al. "Who Calls The Shots? Rethinking Few-Shot Learning for Audio". In: *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). ISSN: 1947-1629. Oct. 2021, pp. 36–40. DOI: 10.1109/WASPAA52581.2021.9632677. URL: https://ieeexplore.ieee.org/document/9632677 (visited on 03/16/2025).

[57] Marco A. Martínez Ramírez et al. "Automatic music mixing with deep learning and out-of-domain data". In: *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*. Ed. by Preeti Rao et al. 2022, pp. 411–418. URL: https://archives.ismir.net/ismir2022/paper/000049.pdf (visited on 04/01/2025).

[58] Thomas Schmitz. "Nonlinear modeling of the guitar signal chain enabling its real-time emulation". PhD thesis. ULiège - Université de Liège, Liège, Belgium, 2019. URL: http://pc-dsp.montefiore.ulg.ac.be/ (visited on 03/18/2025).

[59] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Dec. 3, 2019. DOI: 10.48550/arXiv.1912.01703. arXiv: 1912.01703[cs]. URL: http://arxiv.org/abs/1912.01703 (visited on 04/04/2025).

[60] Omry Yadan. *Hydra - A Framework for Elegantly Configuring Complex Applications*. 2019. URL: https://github.com/facebookresearch/hydra.

[61] William Falcon and Kyunghyun Cho. *A Framework For Contrastive Self-Supervised Learning And Designing A New Approach*. Aug. 31, 2020. DOI: 10.48550/arXiv.2009.00104. arXiv: 2009.00104[cs]. URL: http://arxiv.org/abs/2009.00104 (visited on 04/04/2025).

[62] Ryan Soklaski et al. *Tools and Practices for Responsible AI Engineering*. Jan. 14, 2022. DOI: 10.48550/arXiv.2201.05647. arXiv: 2201.05647[cs]. URL: http://arxiv.org/abs/2201.05647 (visited on 04/04/2025).

[63] Simon Rouard, Francisco Massa, and Alexandre Défossez. “Hybrid Transformers for Music Source Separation”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN: 2379-190X. June 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096956. URL: https://ieeexplore.ieee.org/document/10096956 (visited on 04/20/2025).

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Andrei Guzovski

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Development of a Toolkit for Aggregation and Generation of Real-World Guitar Audio Data", supervised by Uljana Reinsalu
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

12.05.2025

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 - Performance comparisons of experiments performed using RemFx

Table 6. Performance by Normalization Strategy: SI-SDR / ΔSI-SDR (dB) ↑

Effect Removal SI-SDR (dB)		**Test**		
		Loudness (-23)	Loudness (-32)	Peak
Train	Loudness (-23 LUFS)	17.75 (+14.62)	19.42 (+11.90)	11.99 (+10.10)
	Loudness (-32 LUFS)	8.57 (+5.44)	**25.84 (+18.31)**	4.76 (+2.86)
	Peak	15.51 (+12.38)	12.87 (+5.35)	13.63 (+11.74)

Table 7. Performance by Normalization Strategy: MRSTFT / ΔMRSTFT ↓

Effect Removal MRSTFT		**Test**		
		Loudness (-23)	Loudness (-32)	Peak
Train	Loudness (-23 LUFS)	0.43 (-1.80)	0.41 (-0.83)	5.88 (-2.74)
	Loudness (-32 LUFS)	1.11 (-1.13)	**0.22 (-1.01)**	6.32 (-2.30)
	Peak	2.18 (-0.06)	2.32 (+1.09)	0.71 (-7.92)

Table 8. Performance by Data Composition (Acoustic, Electric, Combined): SI-SDR / ΔSI-SDR (dB) ↑

Effect Removal SI-SDR (dB)		**Test**		
		Acoustic	Electric	Combined
Train	Acoustic	**24.06 (+16.28)**	21.40 (+13.88)	22.03 (+14.44)
	Electric	20.58 (+12.80)	25.84 (+18.31)	24.05 (+16.47)
	Combined	23.70 (+15.92)	**26.17 (+18.65)**	**25.24 (+17.65)**

Table 9. Performance by Data Composition (Acoustic, Electric, Combined): MRSTFT / ΔMRSTFT ↓

Effect Removal MRSTFT		**Test**		
		Acoustic	Electric	Combined
Train	Acoustic	**0.20 (-0.82)**	0.42 (-0.82)	0.35 (-0.83)
	Electric	0.31 (-0.71)	0.22 (-1.01)	0.25 (-0.92)
	Combined	0.20 (-0.82)	**0.22 (-1.01)**	**0.22 (-0.96)**

Table 10. Performance by Data Musical Texture (Monophonic, Polyphonic, Combined): SI-SDR / ΔSI-SDR (dB) ↑

Effect Removal SI-SDR (dB)		**Test**		
		Monophonic	Polyphonic	Combined
Train	Monophonic	25.63 (+17.34)	18.90 (+11.31)	19.84 (+12.26)
	Polyphonic	21.30 (+13.00)	**25.36 (+17.77)**	24.83 (+17.24)
	Combined	**26.11 (+17.81)**	24.99 (+17.39)	**25.24 (+17.65)**

Table 11. Performance by Data Musical Texture (Monophonic, Polyphonic, Combined): MRSTFT / ΔMRSTFT ↓

Effect Removal MRSTFT		**Test**		
		Monophonic	Polyphonic	Combined
Train	Monophonic	**0.30 (-0.88)**	0.46 (-0.67)	0.44 (-0.73)
	Polyphonic	0.37 (-0.81)	0.19 (-0.95)	0.22 (-0.95)
	Combined	0.31 (-0.88)	**0.20 (-0.94)**	**0.22 (-0.96)**

Table 12. Performance by Data Aggregation Strategy (Individual vs. Combined): SI-SDR / ΔSI-SDR (dB) ↑

Effect Removal SI-SDR (dB)		**Test**				
		μ Acoustic	Acoustic	μ Electric	Electric	Combined
Train	μ Acoustic	22.16 (+14.63)	23.20 (+15.42)	21.64 (+13.63)	21.43 (+13.91)	21.82 (+14.24)
	Acoustic	22.68 (+15.15)	**24.06 (+16.28)**	22.53 (+14.52)	21.40 (+13.88)	22.03 (+14.44)
	μ Electric	20.42 (+12.89)	19.64 (+11.86)	22.56 (+14.55)	21.76 (+14.24)	21.17 (+13.59)
	Electric	22.03 (+14.50)	20.58 (+12.80)	25.60 (+17.59)	25.84 (+18.31)	24.05 (+16.47)
	Combined	**24.11 (+16.58)**	23.70 (+15.92)	**26.63 (+18.62)**	**26.17 (+18.65)**	**25.24 (+17.65)**

Table 13. Performance by Data Aggregation Strategy (Individual vs. Combined): MRSTFT / ΔMRSTFT ↓

Effect Removal MRSTFT		**Test**				
		μ Acoustic	Acoustic	μ Electric	Electric	Combined
Train	μ Acoustic	0.34 (-0.76)	0.25 (-0.78)	0.37 (-0.78)	0.41 (-0.82)	0.36 (-0.82)
	Acoustic	0.31 (-0.79)	**0.20 (-0.82)**	0.36 (-0.79)	0.42 (-0.82)	0.35 (-0.83)
	μ Electric	0.41 (-0.69)	0.41 (-0.61)	0.41 (-0.75)	0.42 (-0.82)	0.41 (-0.76)
	Electric	0.29 (-0.81)	0.31 (-0.71)	0.24 (-0.91)	0.22 (-1.01)	0.25 (-0.92)
	Combined	**0.23 (-0.88)**	0.20 (-0.82)	**0.21 (-0.94)**	**0.22 (-1.01)**	**0.22 (-0.96)**

Table 14. Performance by Augmentation Strategy (Baseline vs. Advanced): SI-SDR / ΔSI-SDR (dB) ↑

Effect Removal SI-SDR (dB)		**Test**	
		μ Advanced (NN)	μ Baseline (Tanh)
Train	μ Advanced (NN)	**11.28 (+23.97)**	-1.09 (-8.72)
	μ Baseline (Tanh)	-11.88 (+0.81)	**24.53 (+16.90)**

Table 15. Performance by Augmentation Strategy (Baseline vs. Advanced): MRSTFT / ΔMRSTFT ↓

Effect Removal MRSTFT		**Test**	
		μ Advanced (NN)	μ Baseline (Tanh)
Train	μ Advanced (NN)	**0.75 (-0.86)**	1.07 (-0.07)
	μ Baseline (Tanh)	1.60 (-0.02)	**0.26 (-0.89)**