공학석사 학위논문

도메인 적대적 학습을 통해 학습된 강건한 음악 표현을 사용한 멀티미디어 콘텐츠에서의 음악 자동 태깅

Music Auto-tagging in Multimedia Content using Robust Music Representation Learned via Domain Adversarial Training

2023 년 8 월

서울대학교 융합과학기술대학원

지능정보융합학과

정 해 선

공학석사 학위논문

도메인 적대적 학습을 통해 학습된 강건한 음악 표현을 사용한 멀티미디어 콘텐츠에서의 음악 자동 태깅

Music Auto-tagging in Multimedia Content using Robust Music Representation Learned via Domain Adversarial Training

2023 년 8 월

서울대학교 융합과학기술대학원

지능정보융합학과

정 해 선

도메인 적대적 학습을 통해 학습된 강건한 음악 표현을 사용한 멀티미디어 콘텐츠에서의 음악 자동 태깅

Music Auto-tagging in Multimedia Content using
Robust Music Representation Learned via
Domain Adversarial Training

지도교수 이교구

이 논문을 공학석사 학위논문으로 제출함
2023 년 8 월

서울대학교 융합과학기술대학원
지능정보융합학과 지능정보융합학전공
정해선

정해선의 석사 학위논문을 인준함
2023 년 8 월

위 원 장 이 원 종 (인)

부위원장 이 교 구 (인)

위 원 곽 노 준 (인)

Abstract

Music auto-tagging plays a vital role in music discovery and recommendation by assigning relevant tags or labels to music tracks. However, existing models in the field of Music Information Retrieval (MIR) often struggle to maintain high performance when faced with real-world noise, such as environmental noise and speech commonly found in multimedia content like YouTube videos.

In this research, we draw inspiration from previous studies focused on speech-related tasks and propose a novel approach to improve the performance of music auto-tagging on noisy sources. Our method incorporates Domain Adversarial Training (DAT) into the music domain, enabling the learning of robust music representations that are resilient to the presence of noise. Unlike previous speech-based research, which typically involves a pretraining phase for the feature extractor followed by the DAT phase, our approach includes an additional pretraining phase specifically designed for the domain classifier. By this additional training phase, the domain classifier effectively distinguishes between clean and noisy music sources, enhancing the feature extractor's ability not to distinguish between clean and noisy music.

Furthermore, we introduce the concept of creating noisy music source data with varying signal-to-noise ratios. By exposing the model to different levels of noise, we promote better generalization across diverse environmental conditions. This enables the model to adapt to a wide range of real-world scenarios and perform robust music auto-tagging.

Our proposed network architecture demonstrates exceptional performance in music auto-tagging tasks, leveraging the power of robust music representations even on noise types that were not encountered during the training phase. This highlights the model's ability to generalize well to unseen noise sources, further enhancing its effectiveness in real-world applications.

Through this research, we address the limitations of existing music auto-tagging models and present a novel approach that significantly improves performance in the presence of noise. The findings of this study contribute to the advancement of music processing applications, enabling more accurate and reliable music classification and organization in various industries.

Keywords: Robust Music Representation, Music Auto-tagging, Domain Adversarial Training

Student Number: 2021-21802

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Music Auto-tagging refers to the automated process of assigning relevant semantic tags to musical pieces, typically driven by machine learning algorithms, to facilitate effective music information retrieval, personalization, and recommendation systems.

The deployment of music auto-tagging mostly employs clean, pure music tracks, primarily serving the role of facilitating personalized recommendations within music-streaming services such as Spotify[1] (figure 1.1). These services utilize extensive metadata associated with each track to craft a rich and tailored user experience. This metadata, derived from clean musical sources, provides a comprehensive understanding of the inherent attributes of each track, thus enabling a more accurate alignment with individual user preferences.

Moreover, music auto-tagging plays a crucial role in enhancing music search and exploration on video-streaming platforms like YouTube[2]. By assigning relevant se-

[1] https://open.spotify.com/

[2] https://www.youtube.com/results?search_query=live+performance

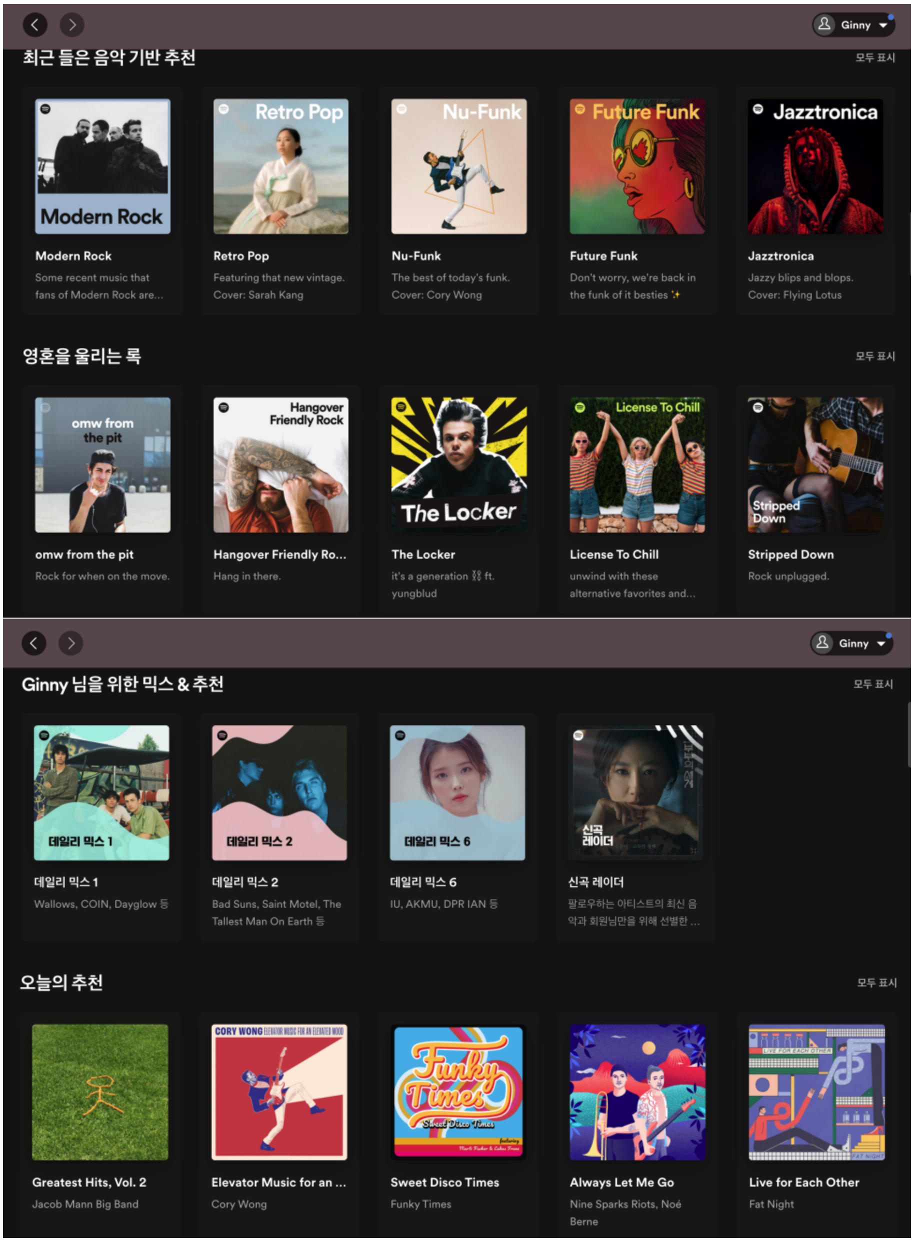
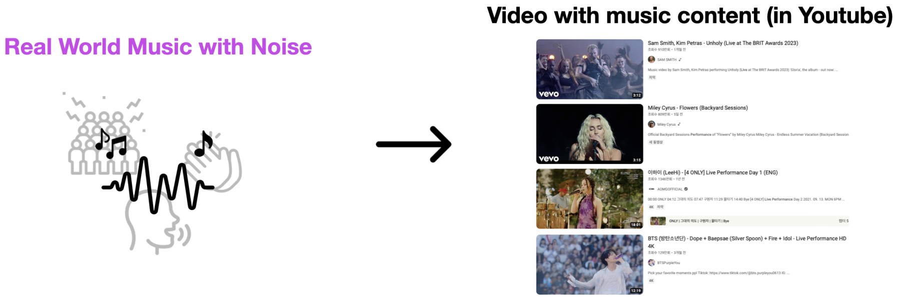


Figure 1.1: Personalized mix and recommendation of music for the user in music-streaming service, Spotify.

mantic tags to music content, users can easily search for specific genres, artists, moods, or themes, enhancing the overall accessibility and discoverability of music content.


Real World Music with Noise
Video with music content (in Youtube)
vevo
3:12
vevo
3:15
18:01
12:19

Sam Smith, Kim Petras - Unholy (Live at The BRIT Awards 2023)
조회수 910만회 • 1개월 전
SAM SMITH
Music video by Sam Smith, Kim Petras performing Unholy (Live at The BRIT Awards 2023) 'Gloria', the album - out now: ...
자막

Miley Cyrus - Flowers (Backyard Sessions)
조회수 809만회 • 5일 전
Miley Cyrus
Official Backyard Sessions **Performance** of "Flowers" by Miley Cyrus Miley Cyrus - Endless Summer Vacation (Backyard Session
새 동영상

이하이 (LeeHi) - [4 ONLY] Live Performance Day 1 (ENG)
조회수 1346만회 • 1년 전
AOMGOFFICIAL
00:00 ONLY 04:12 그대의 의도 07:47 구원자 11:29 물타기 14:40 Bye [4 ONLY] **Live Performance** Day 2 2021. 09. 13. MON 6PM ...
4K 자막
ONLY | 그대의 의도 | 구원자 | 물타기 | Bye 챕터 5

BTS (방탄소년단) - Dope + Baepsae (Silver Spoon) + Fire + Idol - Live Performance HD 4K
조회수 129만회 • 3개월 전
BTSPurpleYou
Pick your favorite moments ppl Tiktok: https://www.tiktok.com/@bts.purpleyou0613 IG: ...
4K

Figure 1.2: Music-related video content such as recordings of live concerts or live performances in video-streaming service, YouTube.

However, it is important to note that there are not enough tags for effective searching and exploring in video-streaming services. The sheer volume and diversity of music-related content uploaded daily make it challenging to rely solely on existing tags. Hence, the development of advanced music auto-tagging techniques becomes imperative to improve the accuracy and efficiency of music search and exploration, enabling users to discover relevant music content more easily.

Furthermore, the music content in video-streaming services contains not just clean music, but also real-world sounds such as crowd noises and applause along with the music (figure 1.2). This presents a unique challenge: the lack of comparable tagged datasets for these ”noisy” music sources limits the effectiveness of traditional auto-tagging mechanisms. AI models trained predominantly on clean music sources struggle to recognize and accurately classify these more complex, noise-infused tracks. Consequently, it becomes imperative to develop and adapt auto-tagging mechanisms capable of accurately labeling these diverse video sources.

The current situation highlights the necessity for novel strategies in music auto-tagging. These advancements would seek to bridge the divide between the metadata tagging used in music-streaming services and the largely untapped potential of complex music content found on video-streaming platforms. This, in turn, would contribute to a richer and more personalized user experience across various platforms.

1.2 Problem Definition

In this section, we explore the specific issues that arise within the context of music auto-tagging. One critical problem stems from the innate limitations of models primarily trained on clean music tracks. When confronted with noisy music input, these models encounter difficulties isolating the musical elements within a composite auditory environment comprising music and extraneous real-world noises. The challenge is heightened in music auto-tagging tasks, where accurate identification and extraction of musical features are crucial.

Existing models encounter a significant obstacle: they must maintain consistent feature extraction irrespective of whether the input originates from a clean or a noisy version of the same music. Due to the prevalent use of clean music sources in training, the models' ability to achieve this consistency is limited. The discrepancy in feature extraction between clean and noisy versions of the same music hinders the models' capacity to tag accurately and effectively.

We propose a novel approach to address this issue: the development of a music-tagged dataset containing both clean and noisy versions of identical music tracks. This methodology allows models to be trained to recognize music amidst the noise and extract consistent features regardless of the input's quality. This proposed solution aims to enhance the model's ability to generalize and perform accurately across diverse audio conditions.

The introduction of such a comprehensive training approach could represent a significant leap forward in the domain of music auto-tagging. However, this novel approach presents its challenges, including the creation and validation of a suitable dataset. The path ahead demands a shift in model training and evaluation paradigms and a more inclusive approach to the complex and diverse reality of real-world music content.

Chapter 2

Background

This chapter provides an overview of the basic concepts and related works to understand the thesis. It covers music representation learning, robust music representation learning, music auto-tagging, domain adversarial training, prior research on music auto-tagging, and general methods for music enhancement, and describes the baseline research that inspired this thesis. By exploring these topics, this chapter establishes the foundation for the subsequent chapters and highlights the significance of the research in advancing music auto-tagging and robust music representation learning.

2.1 Basic Concepts

2.1.1 Music Representation Learning

Music representation learning is a challenging task within the field of machine learning, aiming to develop algorithms that capture the complex and multi-dimensional characteristics of music. It involves transforming raw audio or symbolic musical data into a format that is easily interpretable and usable by machine learning models. The main objective is to create efficient and compact representations that preserve essen-

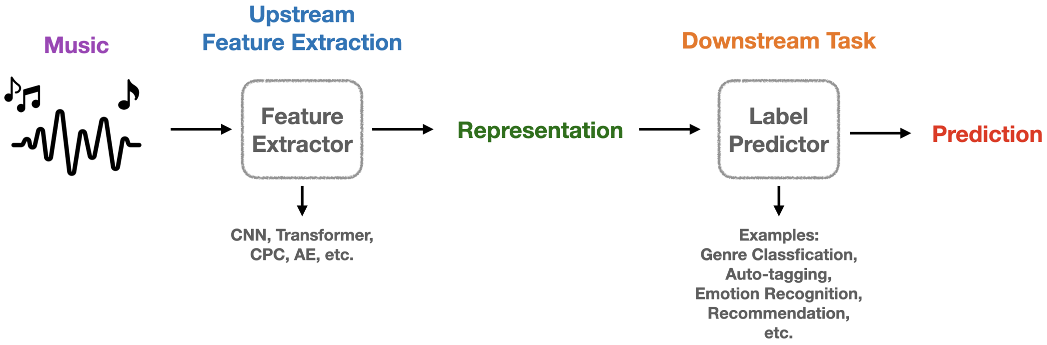


Figure 2.1: Music Representation Learning Framework: the stages of feature extraction and downstream task prediction.

tial musical features, enabling effective analysis, synthesis, and manipulation of music.

Within the context of music information retrieval (MIR), representation learning has played a crucial role in transforming raw music data into a more accessible and manipulable form.

$$X = [x_1, x_2, ..., x_n], \qquad x_i \in \mathbb{R}^r \tag{2.1}$$

$$Mel\,Spectrogram = log(M * |S|), \quad S = STFT(X) \tag{2.2}$$

The raw music audio input X can be discretized into individual samples x_i through a specific sampling rate, serving as the input for the feature extraction process. However, as the size of the discretized samples is large, raw audio can be transformed into a mel-spectrogram, which is a 2-dimensional representation of the frequencies present in an audio signal, emphasizing the human perception of pitch. A shown in equation 2.2, Mel-spectrogram can be computed by applying a mel filterbank to the power spectrum of the audio signal, where each filterbank channel captures a specific frequency range based on the mel scale. The function $STFT$ denotes short-time fourier transform, M denotes mel-filterbank matrix, and $*$ denotes matrix multiplication.

$$Z = [z_1, z_2, ..., z_m], \qquad z_i \in \mathbb{R}^e \tag{2.3}$$

$$f : \mathbb{R}^r \to \mathbb{R}^e, \qquad e \ll r \tag{2.4}$$

Feature extraction aims to map these input samples to a vector representation Z residing in a lower-dimensional embedding space compared to the original input. The transformative nature of Feature Extractor (FE) enables more efficient and effective music representation learning.

The selection of specific techniques and algorithms for FE depends on the desired properties of the resulting embedded representation. Choosing an appropriate FE approach is critical for achieving high-quality and discriminative music representations, which significantly impact downstream tasks such as auto-tagging, genre classification, and recommendation systems.

Ongoing research in music representation learning aims not only to improve the quality of representations but also to gain a deeper understanding of the captured musical elements, enabling more interpretable manipulation. Additionally, the development of unsupervised and self-supervised learning strategies has gained significant interest, as they offer the potential to learn powerful representations from unlabeled data, such as audio recordings and musical scores, which are abundantly available.

2.1.2 Robust Music Representation Learning

In this section, the focus shifts towards robust music representation learning, specifically addressing the challenge of noisy music inputs. While the previous section discussed music representation learning in general, this section delves into the framework

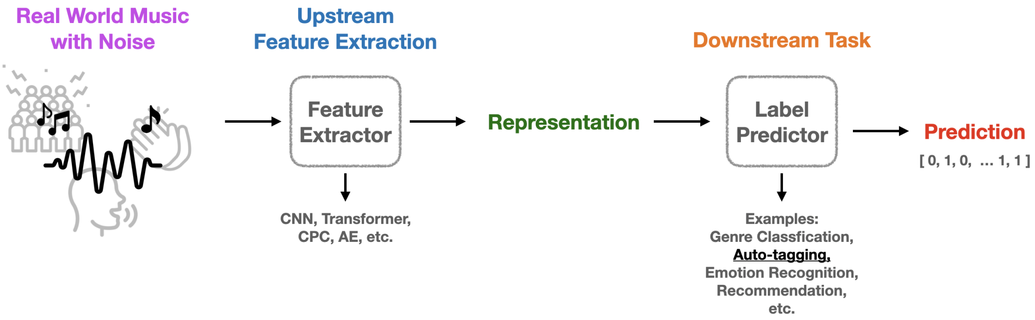


Figure 2.2: Robust Music Representation Learning Framework: specific example of downstream tasks, music auto-tagging with multi-labels.

designed to handle the complexities introduced by noisy music.

The goal of robust music representation learning is to extract representations that are resilient to the presence of additional real-world sounds, such as crowd noises, applause, speech, and other acoustic interferences while preserving the essential musical features. The framework incorporates techniques to ensure that the extracted representations remain consistent regardless of whether the input is clean or noisy music, as long as the musical content remains the same.

The framework builds upon the foundations of music representation learning, leveraging techniques of existing deep learning models. However, it introduces modifications and adaptations to these methods to enhance their robustness to noise and enable consistent feature extraction across varying audio conditions. It aims to disentangle the musical components from the complex auditory environment and extract robust representations that capture the underlying musical attributes.

To address the challenges posed by noisy music inputs, our research aims to develop a robust framework for music auto-tagging in the context of multimedia content. This involves employing advanced signal processing algorithms and deep learning architectures, while also considering the unique challenges presented by noisy music inputs. Instead of relying on complex music enhancement models with a larger number of parameters, our approach focuses on placing both clean and noisy music in the same embedding space. By tailoring the model to handle these challenges, we aim to achieve robust music representation learning with more efficient parameter usage.

2.1.3 Music Auto-tagging

Music auto-tagging is a crucial task in the field of MIR, focusing on automatically assigning relevant tags or labels to music tracks. These tags encompass a broad array of musical characteristics, such as the genre, mood, instrumentation, and other semantic aspects that describe the music content. This form of metadata can facilitate music recommendation, music search, playlist generation, and music content organization, among other applications.

Traditionally, music auto-tagging was performed manually by experts, but this approach is time-consuming and can lack consistency due to the subjective nature of music. Hence, machine learning, and more specifically deep learning, has played an integral role in automating and improving this process.

In the early stages of music auto-tagging, supervised learning methods like Support Vector Machines (SVMs) [1] and k-Nearest Neighbors (k-NN) [2] were popular but faced challenges with the complexity of raw audio data. The advent of deep learning revolutionized music auto-tagging, utilizing Convolutional Neural Networks (CNNs) [3] for spectrogram-based representations and Recurrent Neural Networks (RNNs) [4] for waveform or MIDI-based representations. Recent advancements include attention mechanisms and transformer-based models, enabling focused tag as-

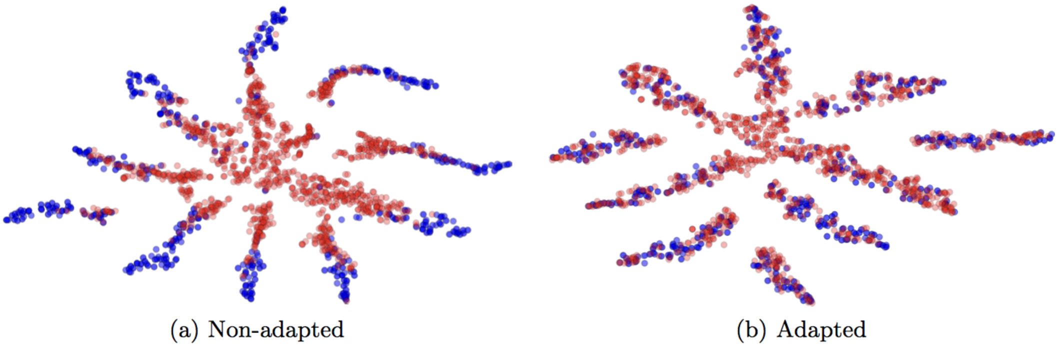


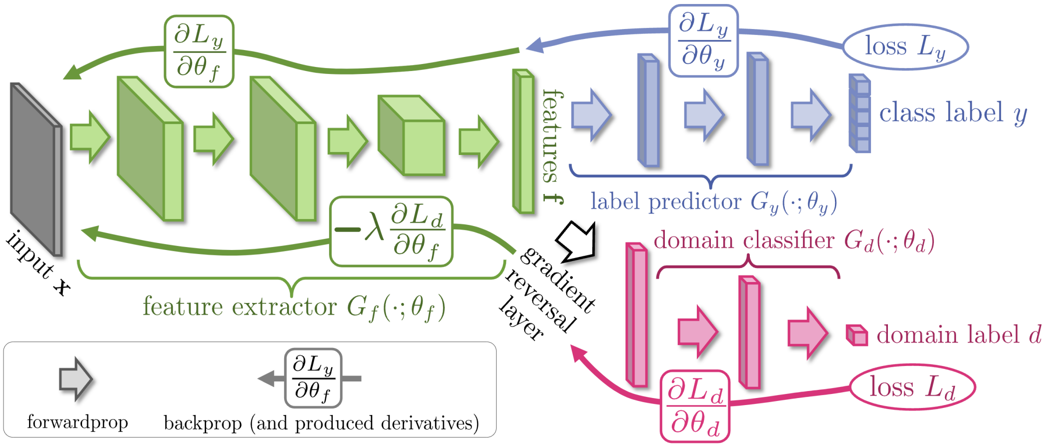
Figure 2.3: The figure of t-SNE visualizations of domain adaptation. *Blue* points correspond to the source domain datapoints, while *red* points correspond to the target domain.

signment by prioritizing relevant parts of music inputs.

Despite the progress made in this field, there are still challenges to overcome. Among these is the handling of the scarcity of labeled data, the problem of tag inconsistency, and the development of models that can provide interpretable reasoning behind their tag assignments. Active research is also being conducted in unsupervised and self-supervised music auto-tagging to address the issue of label scarcity and the high cost of data annotation.

2.1.4 Domain Adversarial Training

Domain Adversarial Training (DAT) [5] is a powerful technique in machine learning that aims to address the challenge of domain shift, where the distribution of data differs between the source domain (used for training) and the target domain (where the model needs to perform well). The goal of domain adversarial training is to learn representations that are domain-invariant, allowing the model to generalize effectively across different domains.



$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
input $\mathbf{x}$
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
loss L_d
$\frac{\partial L_d}{\partial \theta_d}$
forwardprop
backprop (and produced derivatives)

Figure 2.4: The architecture of Domain Adversarial Neural Network (DANN).

In domain adversarial training, three key components are involved: a feature extractor (FE), a domain classifier (DC), and a label predictor (LP) (figure 2.4). The feature extractor learns to extract meaningful representations from the input data, the domain classifier aims to classify the source domain versus the target domain based on the extracted features, and the label predictor is responsible for solving the specific task at hand. These three components are trained simultaneously in an adversarial manner, where the feature extractor and label predictor aim to minimize the task loss, while the domain classifier tries to maximize the domain classification accuracy.

An example of domain adversarial loss function is shown in equation 2.5, where $L(\cdot,\cdot)_y$ and $L(\cdot,\cdot)_d$ is the loss function of LP and DC, θ_f, θ_y, and θ_d are the parameters of the FE, LP, and DC, respectively. Note that i denotes each datapoint. Among the two loss terms of DC, the first one denotes the domain loss of the source domain while the last one denotes the target domain. The parameter λ determines the balance between the two objectives that influence the feature during training. During the training process, the FE and LP aim to maximize the task-specific performance by minimizing the task loss, while the DC aims to maximize the domain classification accuracy by maximizing the domain loss. The domain loss is calculated based on the discrepancy between the predictions of the domain classifier on the source and target domain samples. By jointly training the FE and LP to minimize the task loss while fooling the DC, the model learns to extract features that are robust to domain shift, effectively reducing the domain gap.

$$E(\theta_f, \theta_y, \theta_d) = \frac{1}{n} \sum_{i=1}^{n} L_y^i(\theta_f, \theta_y) - \lambda(\frac{1}{n} \sum_{i=1}^{n} L_d^i(\theta_f, \theta_d) + \frac{1}{n'} \sum_{i=1}^{N} L_d^i(\theta_f, \theta_d)) \tag{2.5}$$

Domain adversarial training has been successfully applied in various domains, including computer vision, natural language processing, and audio processing. It has been used for tasks such as object recognition and sentiment analysis. The technique is particularly useful when labeled data in the target domain is scarce or expensive to obtain, as it allows the model to leverage labeled data from the source domain and adapt it to the target domain.

Overall, domain adversarial training provides a powerful framework for addressing the challenge of domain shift and improving the generalization performance of machine learning models across different domains. By learning domain-invariant representations, the models trained using this technique can effectively transfer knowledge from the source domain to the target domain, enabling improved performance and adaptability in real-world applications.

2.2 Related Work

2.2.1 Music Representation Learning for Auto-tagging

This section provides an overview of prominent deep neural network models that have demonstrated strong performance in the field of music auto-tagging.

Musicnn Musicnn [6] is a convolutional neural network specifically tailored for music auto-tagging tasks. Its architecture comprises three components: a musically motivated CNN for front-end feature extraction, dense layers for mid-end feature extraction, and temporal pooling for classifying 50 tags from the MagnaTagATune dataset [7], as shown in figure 2.5. The framework is designed to extract musically meaningful features that can be utilized for various downstream tasks in accordance with the characteristics of the music. Additionally, the provided pretrained models can be finetuned for transfer learning purposes, enabling their application to other relevant tasks.

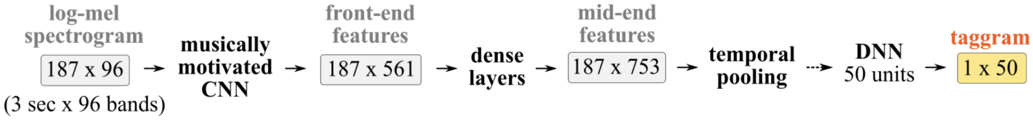


Figure 2.5: The framework of MusiCNN.

When utilizing a vgg-like baseline model [8], the music auto-tagging performance on the MagnaTagATune dataset (MTAT) [7] yielded a ROC-AUC of 90.26 and a PR-AUC of 38.19. Similarly, for the Million Song Dataset (MSD) [9], the achieved ROC-AUC was 87.67, while the PR-AUC reached 28.19.

Jukebox Jukebox [10], a model built upon the Transformer [11] and VQ-VAE [12] architectures, utilizes a hierarchical encoding process to transform raw audio input into a complex representation. Through the VQ-VAE, the input signal is encoded into different-sized vectors, from a sparse representation to a more intricate and detailed representation. This encoding process allows for the effective quantization of the signal into 2048 codes. In addition to the quantized codes, Jukebox incorporates conditional vectors that represent genres or lyrics. By integrating these conditional vectors and leveraging the Transformer model, Jukebox effectively reduces dataset entropy and ensures controllability, resulting in the generation of high-quality pop music audio. By using the extracted feature from the pretrained model with additional strategies [13], this architecture achieved 91.5 on AUC and 41.4 on AP.

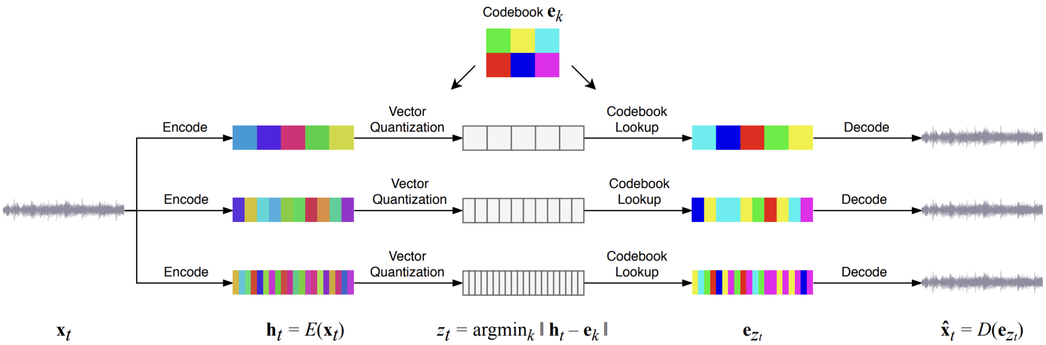


Figure 2.6: The framework of training three separate VQ-VAE of Jukebox.

CLMR CLMR [14] is an approach inspired by SimCLR [15] that leverages contrastive learning to extract musical features. It designates different sections of the same song as positive samples and sections of different songs as negative samples. By applying a comprehensive chain of audio augmentation to randomly selected fragments from raw audio waveforms, positive and negative samples are generated. Using the SampleCNN [16] architecture as the encoder, features are extracted and elaborate representations are learned through contrastive learning. The quality of these representations is evaluated through music classification tasks using datasets like the MTAT dataset and MSD, in which CLMR outperforms other state-of-the-art models.

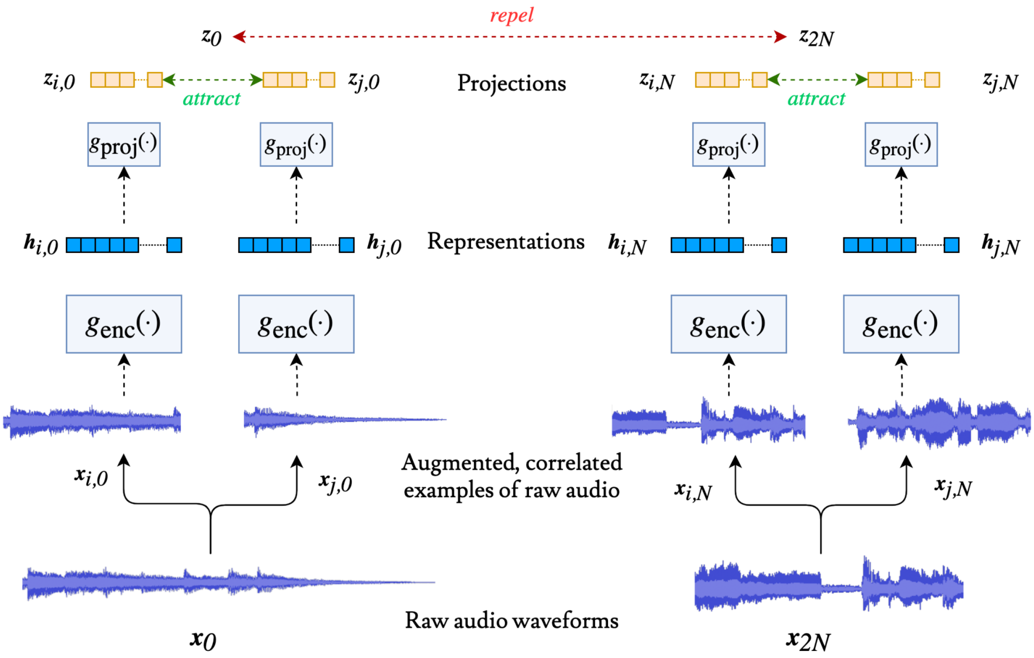


Figure 2.7: The framework of Contrastive Learning for Music Representation (CLMR).

2.2.2 Recent Methods for Music Enhancement

The music enhancement task involves improving the quality of noisy or corrupted music, and the representations generated by these models have the potential to benefit other downstream tasks. While certain architectures employ separate encoder and vocoder components, allowing for the utilization of the encoder's extracted output, other approaches directly address music quality improvement by predicting tokens using Transformer-based models. Considering the objective of enhancing robustness to noise and addressing downstream music-related tasks, these studies offer valuable insights that can contribute to the resolution of the research problem at hand.

Two-stage approach of denoising and enhancement via GAN and DDPM

Kandpal et al. [17] proposed a music enhancement approach for which they employed Mel2Mel [18] and Diffwave [19] models. The Mel2Mel model is utilized to enhance distorted music's mel-spectrogram, while the Diffwave model serves as a vocoder responsible for converting the mel-spectrogram into waveform. The authors explored two training strategies: independent training, where Mel2Mel and Diffwave are trained separately, and joint training, where the models are trained together. In independent training, the advantage lies in the robustness of the Diffwave vocoder to artifacts in the enhanced mel-spectrogram, as it is solely trained on clean mel-spectrograms. Conversely, joint training offers the advantage of achieving a high FAD (Fréchet Audio Distance) [20] score, which is closely related to human perceptual quality.

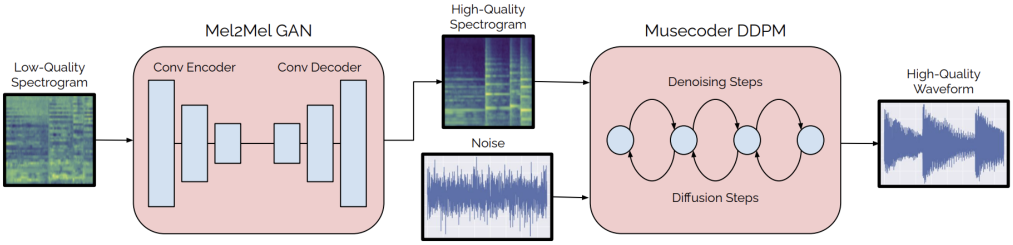


Figure 2.8: Model architecture of Mel2Mel + Diffwave model. This first generates the high-quality mel-spectrogram from the low-quality with the conditional GAN, and then synthesizes high-quality audio waveform from Gaussian noise, conditioned on high-quality mel-spectrogram by using Denoising Diffusion Probabilistic Model (DDPM).

Post-processing approach along with the source separation

Schaffer et al. [21] conducted research on music enhancement by proposing a post-processing model called the *Make it Sound Good (MSG)* post-processor, aimed at enhancing the output of music source separation systems. The study addressed the perceptual shortcomings of state-of-the-art music separation systems, such as the introduction of extraneous noise or the loss of harmonics. The MSG post-processor was applied to both waveform-based and spectrogram-based music source separators, including an unseen separator during training. The analysis of errors produced by source separators revealed that waveform models introduced more high-frequency noise, while spectrogram models lost transients and high-frequency content. Objective measures were introduced to quantify these errors, and the MSG post-processor demonstrated improvements in source reconstruction for both types of errors. Furthermore, subjective evaluations conducted with crowdsourced listeners indicated a preference for MSG post-processed source estimates of bass and drums. It is worth noting that the research by Schaffer et al. acknowledged the limitation of enhancing single-stem outputs of source separation models.

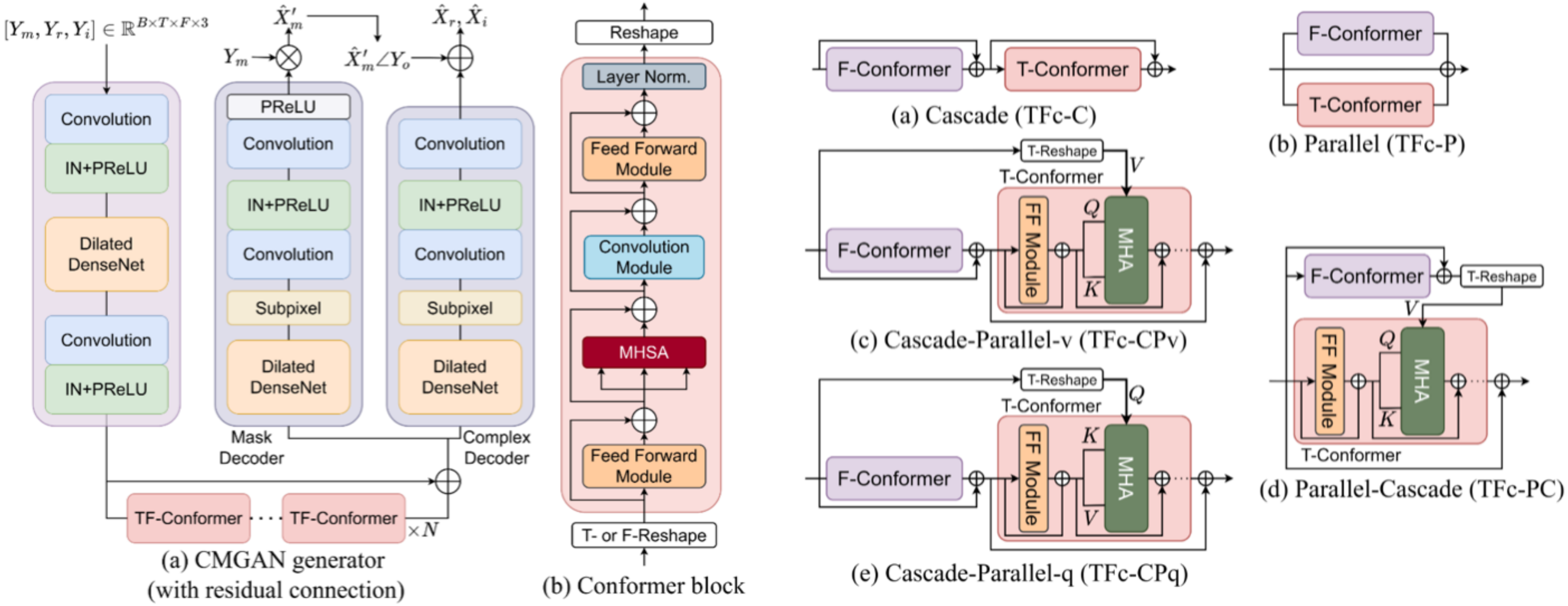
Transformer-based approach

Chae [22] addresses the increasing demand for music enhancement in order to improve the quality of distorted musical recordings. The proposed approach utilizes TF-Conformers, which have demonstrated excellent performance in speech enhancement tasks [23]. The study explores various self-attention techniques of the Conformer model to identify the most effective approach for music enhancement. Experimental results indicate that the proposed model surpasses the state-of-the-art in enhancing single-stem music. Notably, the system also demonstrates the ability to perform general music enhancement with multi-track mixtures, an aspect that previous works have not explored extensively. The methods employed involve TF-Conformer modules based on the encoder and decoders of the CMGAN generator [24], which consists of dilated DenseNet layers, instance normalization, and PReLU activation functions. The proposed TF-Conformers, including Cascade, Parallel, and Cascade-Parallel modules, are further introduced and evaluated. It is noteworthy that these methods have not been previously proposed in TF-self-attention-based models.

2.2.3 Improving Robustness for Speech Representation via Domain Adversarial Training

In the baseline research which inspired this thesis, Huang et al. [25] proposed a novel method using domain adversarial training (DAT) to address the degradation of speech performance caused by various types of distortions. Unlike existing DAT approaches, their method employed a two-stage training process. In the first stage, they utilized a pretrained Hubert model [26] and fine-tuned it on the Librispeech dataset [27] without labels, dividing the dataset into four parts to represent different domains. One part served as the source domain, consisting of clean speech without distortions, while the other three parts represented the target domain with specific distortions: reverb, Gaussian noise, and noise from the *Musan* [28] dataset. This continual training stage aimed to enhance the model's representation and improve its robustness to different types of noise.

In the second stage, the authors utilized the SUPERB [29] framework to conduct experiments on five downstream tasks related to speech. Each task's dataset was divided into source and target domains, with the source domain containing clean audio with known labels, and the target domain consisting of distorted audio with the three aforementioned types of noise. Importantly, the parameters of the FE were trained alongside the DC and LP. Evaluation of the models on unseen distortions demonstrated improved performance compared to the baseline architecture for certain tasks. Notably, the models showed no degradation for the seen noises and, in some cases, even outperformed the fully supervised model.



(a) CMGAN generator (with residual connection)

(b) Conformer block

(a) Cascade (TFc-C)

(b) Parallel (TFc-P)

(c) Cascade-Parallel-v (TFc-CPv)

(d) Parallel-Cascade (TFc-PC)

(e) Cascade-Parallel-q (TFc-CPq)

Figure 2.9: TF-Conformer

Chapter 3

Method

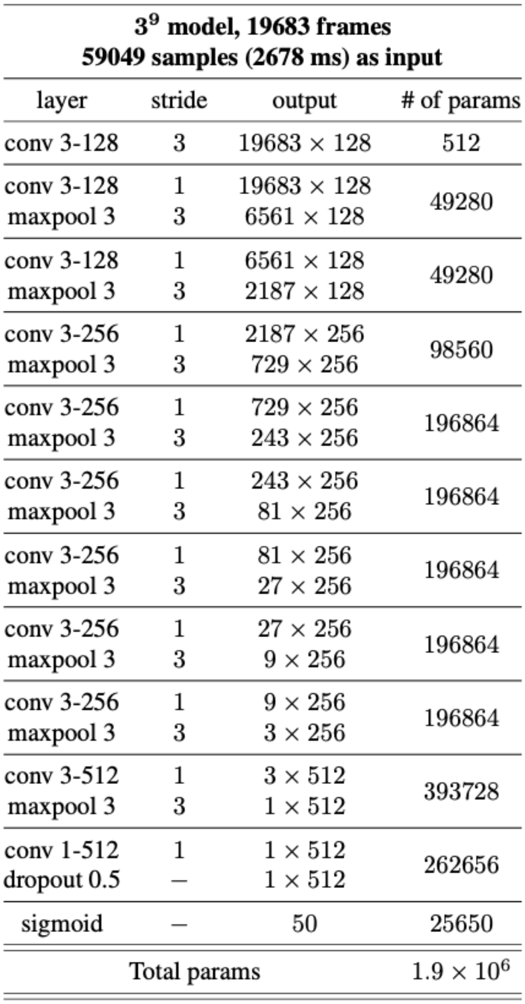
In this chapter, we present a detailed explanation of the proposed architecture, training process, and objective function. These components are based on the foundation laid by previous research, which focused on enhancing the robustness of speech representation as discussed in the preceding section.

3.1 Model Structure

The architecture of the domain adversarial training (DAT) consists of three distinct models. In our implementation, we similarly structured the main components of our architecture into three modules, which will be comprehensively discussed in detail.

3.1.1 Feature Extractor

The FE in this study is designed based on the state-of-the-art CLMR architecture, which has demonstrated exceptional performance in music auto-tagging tasks. CLMR leverages the SimCLR [15] backbone, which employs contrastive learning strategies for representation learning without the need for labeled data. In this research, the SampleCNN architecture (figure 3.1) is utilized as the encoder component of FE. Notably, the last fully-connected layer of SampleCNN is substituted with an identity matrix. The resulting encoder output is then passed through a projector module, consisting of two fully-connected layers with a ReLU activation function in between, to obtain the final feature representation.



3^9 model, 19683 frames
59049 samples (2678 ms) as input

layer	stride	output	# of params
conv 3-128	3	19683 × 128	512
conv 3-128 maxpool 3	1 3	19683 × 128 6561 × 128	49280
conv 3-128 maxpool 3	1 3	6561 × 128 2187 × 128	49280
conv 3-256 maxpool 3	1 3	2187 × 256 729 × 256	98560
conv 3-256 maxpool 3	1 3	729 × 256 243 × 256	196864
conv 3-256 maxpool 3	1 3	243 × 256 81 × 256	196864
conv 3-256 maxpool 3	1 3	81 × 256 27 × 256	196864
conv 3-256 maxpool 3	1 3	27 × 256 9 × 256	196864
conv 3-256 maxpool 3	1 3	9 × 256 3 × 256	196864
conv 3-512 maxpool 3	1 3	3 × 512 1 × 512	393728
conv 1-512 dropout 0.5	1 –	1 × 512 1 × 512	262656
sigmoid	–	50	25650
Total params			1.9×10^6

Table 3.1: The SampleCNN architecture, serving as the feature extractor and encoder within the CLMR framework.

Given that the input sample length of the SampleCNN model is 59,049, it corresponds to approximately 2.7 seconds of audio at a sampling rate of 22,050 Hz. The encoder component of the model produces a 50-dimensional vector as its output, which aligns with the number of tags present in the MTAT dataset. Further elaboration on the MTAT dataset and its relevance will be provided in the subsequent section.

3.1.2 Domain Classifier

The structure of the DC is based on the original DAT research. It takes the output embedding from the FE as input and produces a 1-dimensional vector that classifies whether the embedding originated from the clean source input or the noisy target input. The DC model comprises simple fully-connected layers, accompanied by activation functions and batch normalization, as depicted in Figure 3.1.

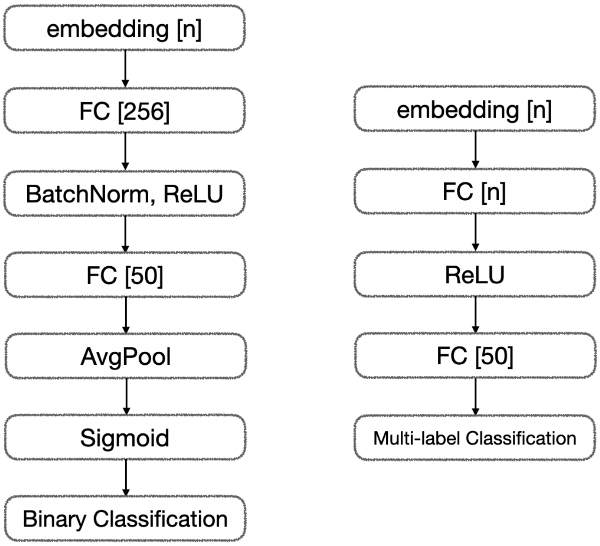


Figure 3.1: The architecture of domain classifier(*left*) and the label predictor(*right*).

3.1.3 Label Predictor

Among the models in the proposed architecture, the LP stands out with its compact structure and minimal number of layers and parameters. As illustrated in Figure 3.1, the LP model takes the output from the FE as input and sequentially processes it through two fully-connected layers, with a ReLU activation function in between. The final output of the LP is a 50-dimensional vector, which corresponds to the classification of the 50 multi-tags in the MTAT dataset.

3.2 Domain Adversarial Training for Clean Source Domain and Noisy Target Domain of Music

The proposed method in this thesis follows a three-stage training approach. In the first stage, the FE is pretrained. The second stage involves the pretraining of the DC while keeping the FE frozen. In the final stage, the FE is finetuned, and simultaneously, the LP is trained.

3.2.1 Pretraining Feature Extractor

In line with previous research [25], the FE was pretrained in this study. However, instead of using pretrained model, we opted to train the CLMR model from scratch using the MTG-Jamendo dataset. Although the pretrained model parameters were provided by the authors, the MTAT dataset, which would be used for downstream tasks, could not be exposed to the model until the final stage. Additionally, full access to the MSD dataset was not available at the time. As a result, the MTG-Jamendo dataset, which is annotated with multi-labels for music auto-tagging, was chosen to allow the model to learn a generalized representation of music audio during the initial pretraining stage. Notably, unlike the previous research, this stage involved training half of the dataset as a clean source and the other half as a noisy target, incorporating a specific number of noise sources. Essentially, the separate stages of pretraining with clean source only and continual training with noisy target were combined into a single stage in this study.

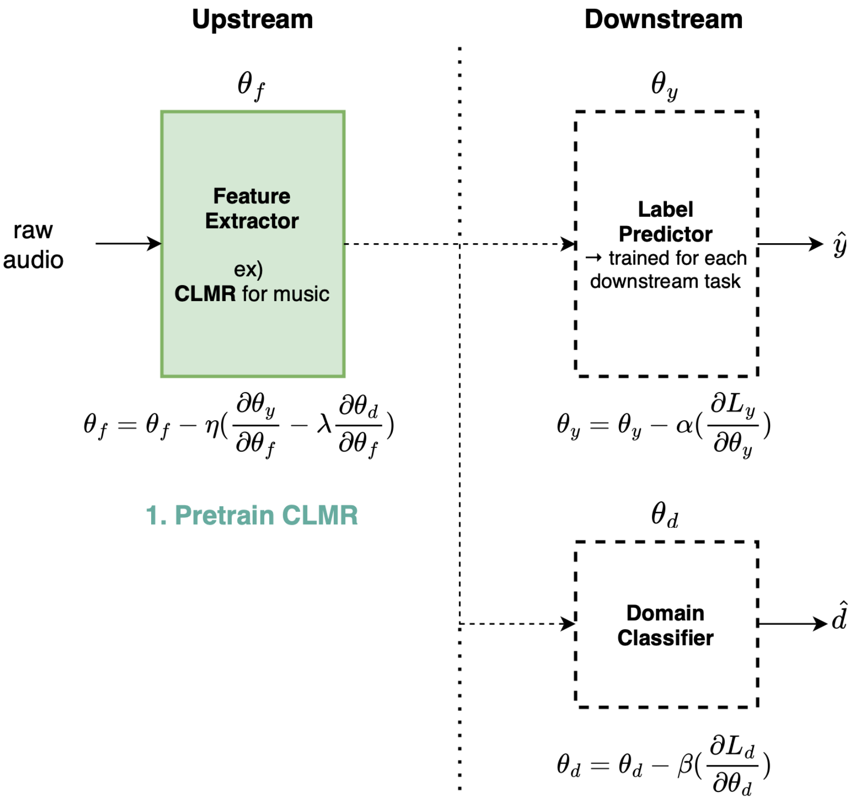

Upstream
Downstream
θ_f
raw audio
Feature Extractor
ex) CLMR for music
$\theta_f = \theta_f - \eta(\frac{\partial\theta_y}{\partial\theta_f} - \lambda\frac{\partial\theta_d}{\partial\theta_f})$
1. Pretrain CLMR
θ_y
Label Predictor
→ trained for each downstream task
$\hat{y}$
$\theta_y = \theta_y - \alpha(\frac{\partial L_y}{\partial\theta_y})$
θ_d
Domain Classifier
$\hat{d}$
$\theta_d = \theta_d - \beta(\frac{\partial L_d}{\partial\theta_d})$

Figure 3.2: The first stage of the proposed method: Pretraining FE with MTG-Jamendo dataset.

3.2.2 Pretraining Domain Classifier

In previous studies on DAT [5] and speech robustness improvement [25], the DC was trained simultaneously with other models. However, in the context of music representation, training DC from scratch in conjunction with FE or LP hindered its performance. Therefore, based on empirical observations, we opted to first pretrain DC while keeping the parameters of pretrained FE frozen to ensure consistent input embeddings. Moreover, we used the MTAT train dataset divided into a clean source domain and a noisy target domain in which configuration will be consistent until the last stage.

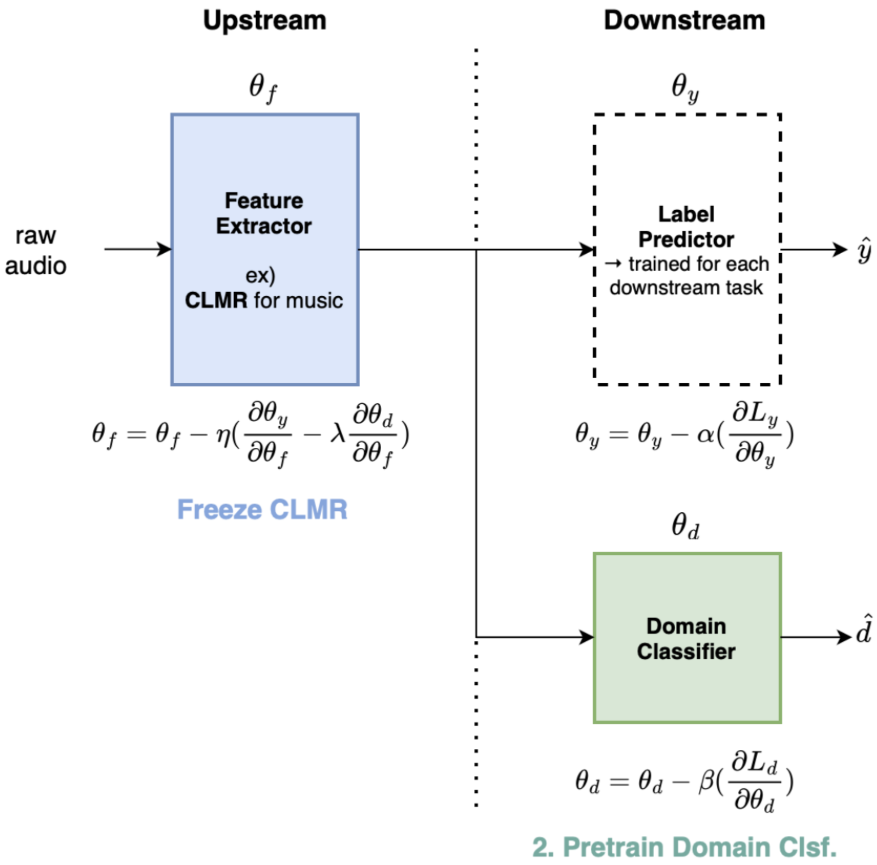

Upstream
Downstream
θ_f
θ_y
raw audio
Feature Extractor
ex) CLMR for music
Label Predictor
→ trained for each downstream task
$\hat{y}$
$\theta_f = \theta_f - \eta(\frac{\partial\theta_y}{\partial\theta_f} - \lambda\frac{\partial\theta_d}{\partial\theta_f})$
$\theta_y = \theta_y - \alpha(\frac{\partial L_y}{\partial\theta_y})$
Freeze CLMR
θ_d
Domain Classifier
$\hat{d}$
$\theta_d = \theta_d - \beta(\frac{\partial L_d}{\partial\theta_d})$
2. Pretrain Domain Clsf.

Figure 3.3: The second stage of proposed method: Pretraining DC with MTAT dataset.

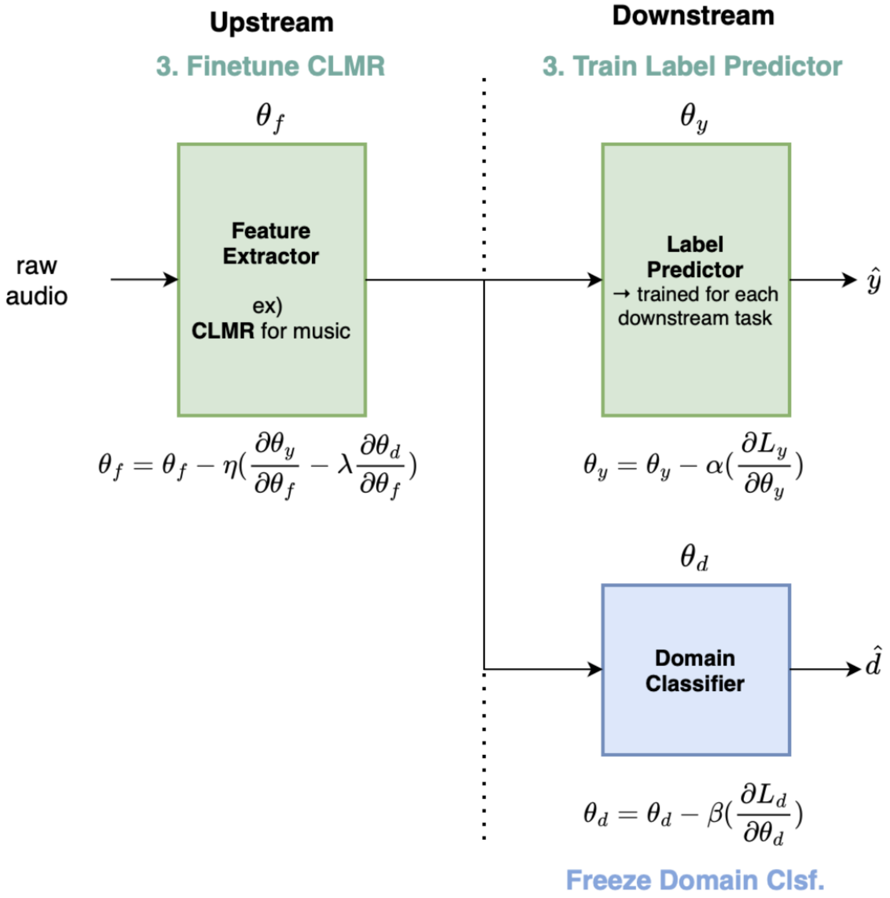

Upstream
3. Finetune CLMR
θ_f
Feature Extractor
ex) CLMR for music
raw audio
$\theta_f = \theta_f - \eta(\frac{\partial\theta_y}{\partial\theta_f} - \lambda\frac{\partial\theta_d}{\partial\theta_f})$
Downstream
3. Train Label Predictor
θ_y
Label Predictor → trained for each downstream task
$\hat{y}$
$\theta_y = \theta_y - \alpha(\frac{\partial L_y}{\partial\theta_y})$
θ_d
Domain Classifier
$\hat{d}$
$\theta_d = \theta_d - \beta(\frac{\partial L_d}{\partial\theta_d})$
Freeze Domain Clsf.

Figure 3.4: The last stage of the proposed method: Training LP and finetuning FE with MTAT dataset.

3.2.3 Domain Adversarial Training for Finetuning Feature Extractor and Training Label Predictor

The final stage of the proposed method focuses on training LP while simultaneously finetuning FE, with the objective of mapping clean and noisy music inputs from the same song to the same embedding space. This is achieved by reversing the gradient of DC and incorporating it into FE, weighted by the hyperparameter λ to balance the scale of domain loss and label loss. Through this approach, FE is trained to disregard the distinction between clean and noisy music inputs, leading LP to result in highly similar classification output for music samples from the same song, regardless of their noise levels.

3.3 Objective Function

At the first stage of pretraining FE, the contrastive learning approach was adapted thus *NT-Xent* loss (normalized temperature-scaled cross-entropy loss) [15] is used for pretraining without labels. The equation of NT-Xent loss is as follows:

$$l_{i,j} = -\log \frac{\exp\left(\operatorname{sim}\left(\mathbf{z}_i, \mathbf{z}_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\operatorname{sim}\left(\mathbf{z}_i, \mathbf{z}_k\right)/\tau\right)} \tag{3.1}$$

Note that the function above is for a positive pair of examples (*i, j*). The function sim(u, v) denotes the dot product between l_2 normalized u and v (i.e. cosine similarity), and $\mathbb{1}_{[k \neq i]}$ is an indicator function evaluating to 1 iff $k \neq i$ and τ denotes a temperature.

In the second stage and the last stage of the proposed method, DC and LP are trained by *BCE* (Binary Cross-Entropy) loss for binary classification for DC and multi-label classification for LP. The equation of BCE loss is described in equation 3.2.

$$BCE(x) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(p) + (1 - y_i) \log(1 - p), \quad p = h(x_i; \theta) \tag{3.2}$$

The total loss function for the DAT stage encompasses the parameter updates for FE and LP, as depicted in equation 3.4. It is important to note that the parameters of DC can be either frozen or finetuned during the final DAT stage, as determined through experimentation. Note that θ_d, θ_y, and θ_f correspond to parameters of DC, LP, and FE, respectively.

$$\theta_d = \theta_d - \beta(\frac{\partial L_d}{\partial \theta_d}) \quad , \quad \theta_y = \theta_y - \alpha(\frac{\partial L_y}{\partial \theta_y}) \tag{3.3}$$

$$\theta_f = \theta_f - \eta(\frac{\partial \theta_y}{\partial \theta_f} - \lambda \frac{\partial \theta_d}{\partial \theta_f}) \tag{3.4}$$

Chapter 4

Experiment

This chapter provides an overview of the music and noise datasets employed in the experiment, along with their respective data configurations. Subsequently, the section presents the implementation details, settings, and architectures of both the baseline and the fully supervised models. Lastly, comprehensive descriptions are provided for the evaluation metrics, settings, and the subsequent results obtained from the evaluation.

4.1 Dataset

4.1.1 Music Dataset

For the music dataset, the *MTG-Jamendo* dataset [30] was utilized to pretrain the FE, while the *MagnaTagATune* dataset [7] was employed for pretraining the DC and training the LP during the DAT stage for the downstream tasks. The FE was specifically trained to exhibit generalization capabilities for both clean source and noisy target musical representations. To achieve this, the FE was pretrained using the MTG-Jamendo dataset, which differs from the datasets utilized for the downstream tasks.

MTG-Jamendo The MTG-Jamendo dataset comprises a total of 56,639 full audio tracks, with a median duration of 224 seconds per track, amounting to approximately 3,777 hours of audio. The dataset covers a range of genres, instruments, and mood/theme categories, encompassing a total of 195 tags. To facilitate specific research requirements, subsets of the dataset containing a specific set of tags have been created, and these subsets are further divided into train, validation, and test splits.

Considering the size of the full dataset, a filtering process was applied to select relevant tracks. Specifically, two subsets were utilized: one containing 95 genre tags and another containing 59 mood/theme tags. By cross-referencing the provided metadata, a subset of 18,255 tracks that included both genre and mood/theme tags was filtered out. Given that the number of genre and mood/theme tags exceeded that of instrument tags, it was inferred that the remaining filtered tracks likely encompassed a significant portion of the remaining tags within the instrument category. Consequently, the final filtered dataset consisted of 1,108 hours of audio data, featuring 87 genre tags and 56 mood/theme tags.

MagnaTagATune The MagnaTagATune dataset encompasses a collection of 25,863 music clips, each lasting approximately 29 seconds. These clips are derived from a set of 5,223 unique songs, distributed across 445 albums and performed by 230 artists. The dataset exhibits a diverse range of musical genres, including Classical, New Age, Electronica, Rock, Pop, World, Jazz, Blues, Metal, Punk, among others. Each audio clip is associated with a binary annotation vector, comprising 188 tags. These annotations are generated through the TagATune game, an online two-player game where participants listen to audio clips and assign descriptive tags. The game involves players hearing the same or different audio clips and providing corresponding tags. Tags are assigned when there is agreement among multiple players. Examples of tags include ‘singer’, ‘no singer’, ‘violin’, ‘drums’, ‘classical’, and ‘jazz’. To ensure sufficient training data for each tag, evaluation is commonly performed using the top 50 most popular tags. The dataset is divided into 16 parts, with parts 1-12 typically used for training, part 13 for validation, and parts 14-16 for testing purposes.

4.1.2 Noise Dataset

The noise dataset used in this study consisted of two primary sources: *Audioset* [31] and the *Musan* dataset [28]. Audioset, a comprehensive collection of labeled audio segments, was utilized for training purposes across all stages of the experiment. In contrast, the Musan dataset was specifically employed to evaluate the performance of the trained models. Within the Musan dataset, the noise subset was selected for inclusion in the study. This experimental design was devised to assess the models’ ability to effectively handle noise from external datasets, thereby providing valuable insights into the robustness and adaptability of the proposed approach.

Audioset The Audioset is a large-scale dataset of human-labeled 10-second sound clips developed by Google. It contains a diverse collection of audio recordings from YouTube videos. The dataset is designed for audio event detection and classification tasks, providing audio clips labeled with a variety of sound events such as musical instruments, animal sounds, human activities, and environmental sounds. Audioset has been widely used in research and machine learning applications for tasks related to audio analysis, including speech recognition, sound event detection, and audio classification.

We performed manual filtering on the dataset, which originally contained 527 labels, to extract music-related labels. The resulting filtered dataset consisted of 348 unique tags for training and 349 tags for evaluation, ensuring there was no overlap between the two sets. The training set contained 10,382 files, while the evaluation set consisted of 9,860 files. Each file in the dataset had a minimum of 1 tag and a maximum of 10 tags. On average, there were 2.2 tags per file in the training set and 2.3 tags per file in the evaluation set. During the training process, we used half of the evaluation set as a validation set, while the remaining half served as the test set. It is important to note that the noise samples used in the test phase were not present in the training or validation sets.

Musan Musan dataset presents a new corpus comprising music, speech, and noise data, which is suitable for training voice activity detection (VAD) models and music/speech discrimination systems. The dataset includes music from various genres, speech in twelve different languages, and a diverse range of technical and non-technical noises. The corpus is released under a flexible Creative Commons license, enabling redistribution of the original audio. The speech portion of the corpus consists of approximately 60 hours of read speech from Librivox and US government hearings, committees, and debates. The music portion, totaling 42 hours and 31 minutes, is sourced from platforms like Jamendo, Free Music Archive, Incompetech, and HD Classical Music. Additionally, the noise segment contains 929 files, featuring an assortment of technical noises and ambient sounds. The dataset facilitates applications such as VAD for speaker identification and music/speech discrimination on Broadcast news.

4.1.3 Data Configuration

In this section, we present the data configuration employed during each training stage, highlighting the specific music and noise datasets utilized, as well as the division of data into training, validation, and testing sets.

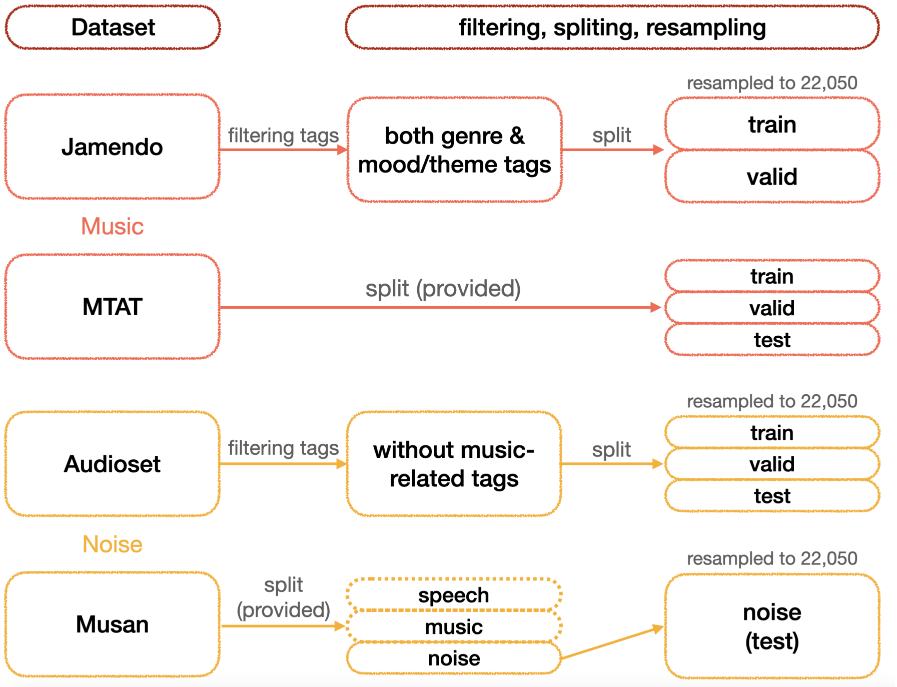


Figure 4.1: Data Preprocessing Pipeline for Music and Noise Datasets.

As outlined in Section 4.1, the experiment utilized a total of four datasets, comprising two datasets for music and two datasets for noise. Each dataset underwent preprocessing steps, including the filtering of relevant tags, division into training, validation, and testing subsets, and resampling to a frequency of 22,050Hz, as shown in figure 4.1.

During the initial stage of FE pretraining, the training subset of the Jamendo dataset, in conjunction with the training subset of Audioset, were utilized (figure 4.2). The data was partitioned into two domains based on data indices: the source domain, comprising clean data, and the target domain, encompassing music data augmented with noise to yield noisy data. For validation purposes, the entire Jamendo validation subset was employed for both the source and target domains. Notably, the target domain was combined with the validation subset of the Audioset.

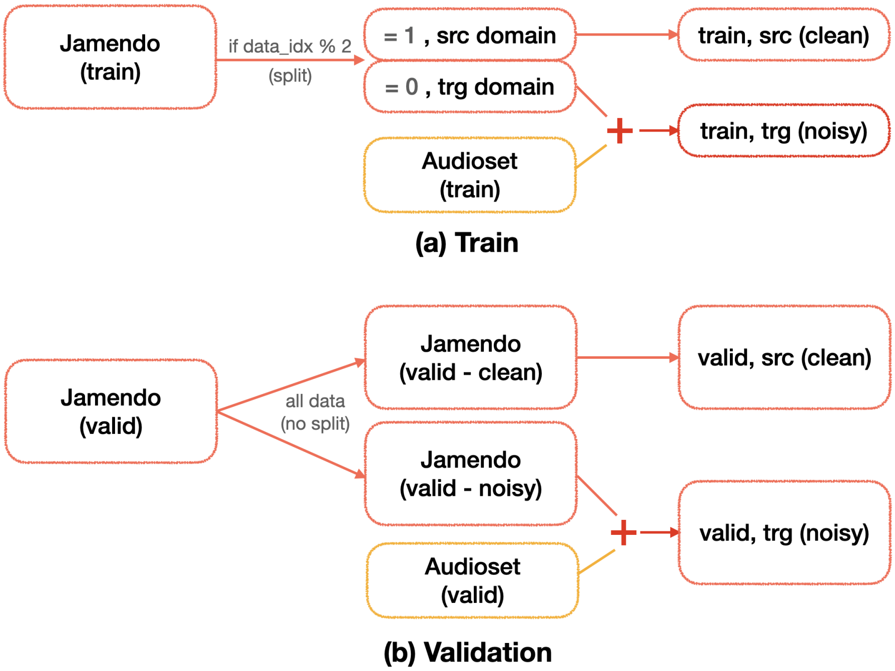
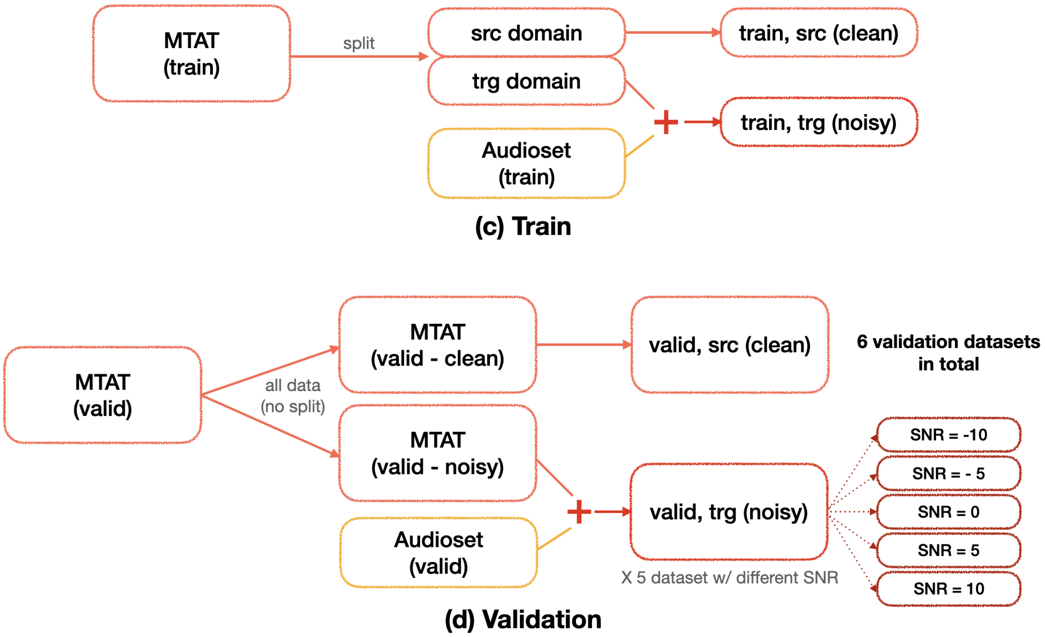


Figure 4.2: Data configuration during the initial stage of FE pretraining.

During the second stage of DC pretraining and the final stage of DAT, the training subset of the MTAT dataset, in conjunction with the training subset of Audioset, was employed. Prior to the experiment, the MTAT training subset was divided into a clean source domain and a noisy target domain, with the noisy data generated by mixing the target domain with noise from the training subset of Audioset. During the training process, the range of SNR for the noisy target data was set to [-10, 10]. Various learning strategies suitable for each specific training stage were employed. For validation purposes, the entire MTAT validation subset was utilized, encompassing six different validation configurations. The first configuration consisted of the clean source domain without any noise, while the remaining five configurations represented the noisy target domain with a varying signal-to-noise ratio (SNR) settings (figure 4.3).



(c) Train

(d) Validation

Figure 4.3: Data configuration for the second stage of DC pretraining and final stage of DAT, with MTAT dataset and Audioset. Training: Clean source and noisy target domains. Noisy target domains are trained with the SNR range of [-10, 10] with the given learning strategies. Validation: Clean source and five noisy target configurations with varying SNR of [-10, -5, 0, 5, 10], respectively.

During the test phase of the experiment, two noise datasets were employed to evaluate the performance of the model. The first dataset was the internal dataset, consisting of the test subset of Audioset. The second dataset was the external dataset, encompassing the complete noise subset of the Musan dataset. The test set for the clean source domain comprised the full test subset of the MTAT dataset. As for the noisy target domain, the test set retained the same signal-to-noise ratio (SNR) configurations utilized during the validation process (figure 4.4).

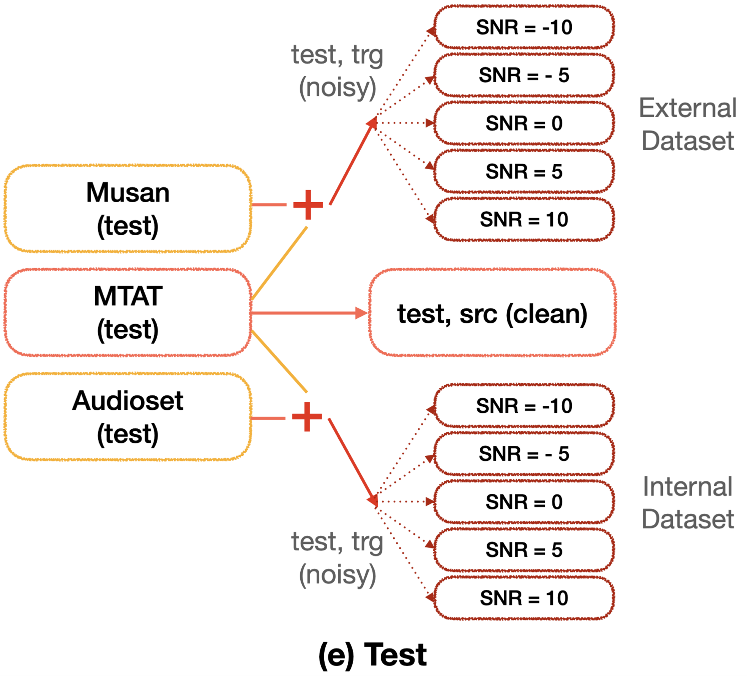


Figure 4.4: Data configuration for the test phase, including internal and external noise datasets and consistent SNR settings for the noisy target domain.

4.2 Implementation Detail

As described in Section 3.1, our model comprises three components: FE based on the architecture of CLMR, and two linear classifiers for DC and LP. The FE encoder is built upon the SampleCNN architecture, consisting of 11 convolutional modules with a kernel size of 3, a stride size of 1, and a padding size of 1. The hidden layers of the FE encoder have dimensions of 128, 256, and 512, respectively. The final fully-connected layer outputs a 50-dimensional vector, corresponding to the number of tags in the MTAT dataset.

Our model was trained on a single NVIDIA 3090 GPU with 24GB of memory. The training process lasted for 800 epochs, with a batch size of 48 samples in the final stage. The training duration for mixing two noises from the noise dataset was approximately 16 hours. The training time varied depending on the number of noises mixed with the clean music source. In this study, we experimented with mixing 1, 2, and 4 noises. The Adam [32] optimizer was utilized with an initial learning rate of 0.0003 for pretraining FE and 0.0001 for all models during the second and final stage of training.

The training process of the baseline model reduces into 2 stages as in this architecture, DC is excluded. The first stage is as same as the proposed setting, but for the final stage, the data configuration and the concerned loss values differ.

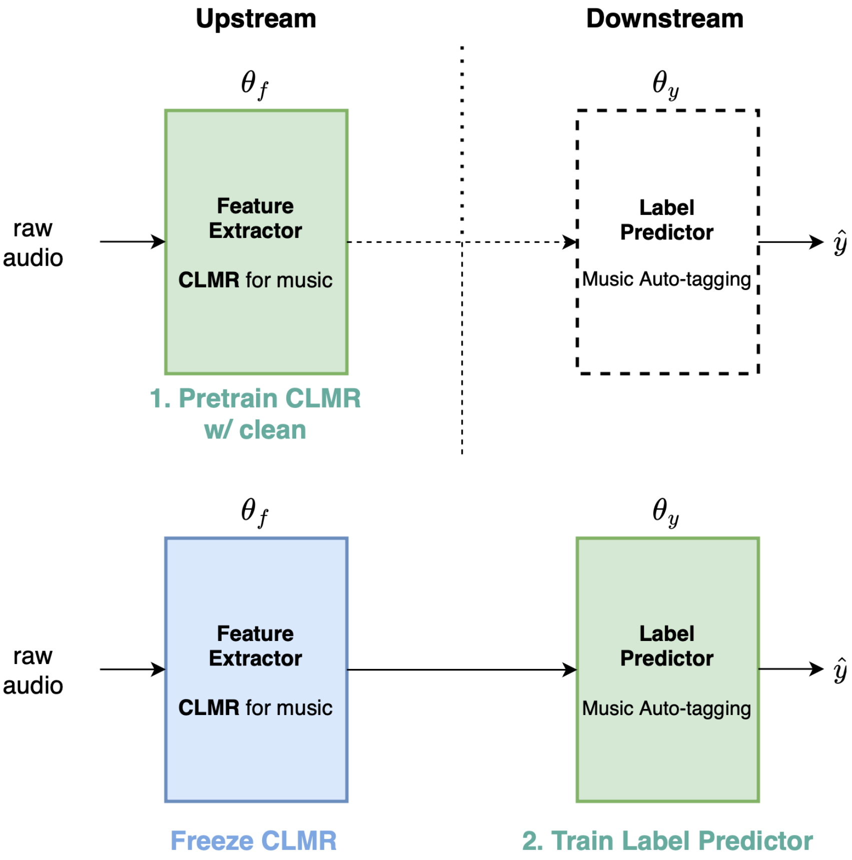


Figure 4.5: The training stages of baseline and fully supervised models.

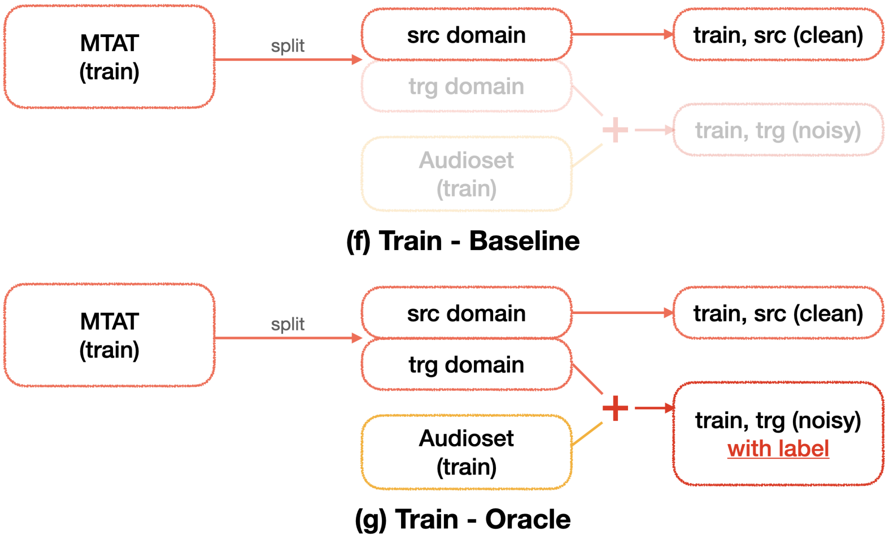


Figure 4.6: The data configuration of baseline and fully supervised models.

The total loss function varies depending on the architecture being trained. The baseline model is solely trained with the source domain loss, while the fully supervised model is trained with both the source and target domain loss. Consequently, the labels for the noisy target domain are provided during training for the fully supervised model. In contrast, the proposed method is trained with both the label and domain loss for the source domain, while only the domain loss is utilized for the target domain where the labels are not available. The domain loss is calculated using BCE loss to classify whether the input embedding of the DC model originated from the clean source domain or the noisy target domain. Additionally, the label loss is also computed using BCE to perform multi-label classification for the 50 tags in the MTAT dataset.

Model \ Loss	Source (clean)		Target (noisy)	
	label	domain	label	domain
Baseline	o	x	x	x
Fully Supervised	o	x	o	x
Proposed	o	o	x	o

Table 4.1: Comparison of loss functions for different models. The baseline model is trained with source label loss only, while the fully supervised model incorporates both source and target label losses. The proposed method utilizes both label and domain losses for the source domain, and only the domain loss for the target domain. ”o” represents the presence of the loss and ”x” indicates the absence of the loss.

4.3 Hyperparameter Settings and Learning Strategies

During the final stage of training, various hyperparameters and learning strategies were explored to optimize performance. Firstly, the weight of the domain loss (λ) was determined empirically to achieve the best performance. A value of 0.05 was found to be optimal.

Secondly, the range of the signal-to-noise ratio (SNR) was adjusted differently for different learning strategies during training. For the *No Learning Strategies* approach, the SNR range was set to [-10, 10]. Random integer values were selected for each noisy sample before mixing it with the clean music. In contrast, when employing the *Easy-to-Hard* (E2H) learning strategy, the initial SNR range was set to [-10, -5] in the second training stage. This range was then gradually increased by 1 every 3 epochs until reaching the maximum value of 10. Similarly, in the final training stage, the SNR range was initially set to [5, 10] and gradually decreased by 1 every 30 epochs until reaching the minimum value of -10. These adjustments were made to ensure that the DC model is progressively exposed to a wider range of SNR conditions.

In addition, an investigation was conducted to compare the effects of incorporating the α value, multiplied by the negation of the gradient of the DC model, into the training process. The calculation method for α was derived from the DANN [5]. This involved considering factors such as the batch index, number of epochs, and length of the dataloader when computing the α value.

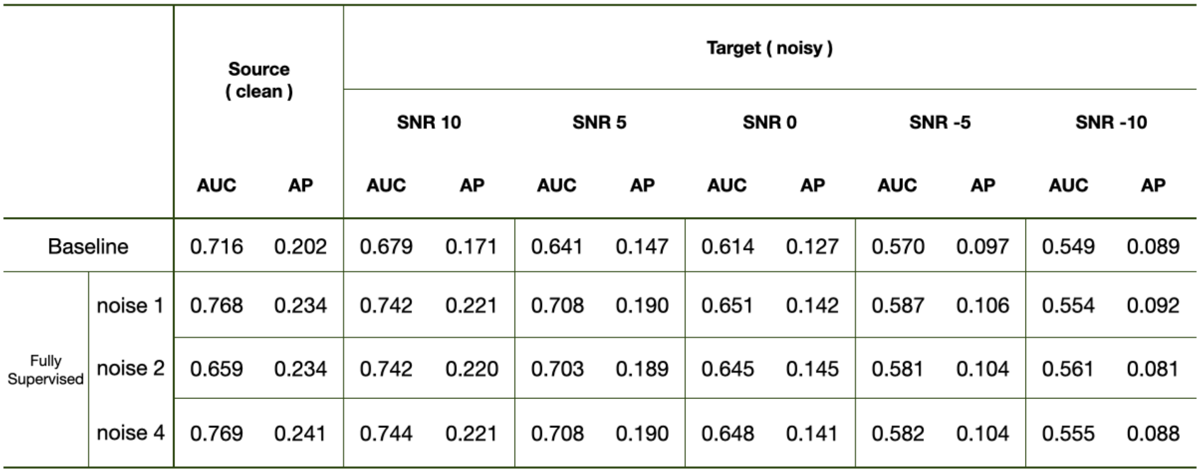
Lastly, when utilizing the pretrained DC model in the final stage, a decision was made regarding whether to fine-tune its parameters or freeze them during training. Interestingly, it was observed that freezing the parameters of the DC model yielded better results in evaluation metrics such as Area Under the Curve (AUC) or Average Precision (AP), despite the finetuned model potentially exhibiting higher validation accuracy and lower loss.

4.4 Results

This section presents a comprehensive comparison and discussion of the evaluation metrics and results for the Baseline, Fully supervised, and Proposed models. The analysis focuses on three key aspects: the model architecture, the number of noises utilized for synthesizing the target domain data, and the tuning of hyperparameters combined with the selection of learning strategies.

In terms of the model architecture, a comparison is made between the Baseline model, trained solely with the source data and tag label loss, the fully supervised model trained with both source and domain data along with tag labels, and the Proposed model, which incorporates source tag labels and domain labels for both source and target domains, excluding target tag labels. Furthermore, the evaluation explores the impact of using different numbers of noises (1, 2, or 4) for synthesizing the target domain data, with a fixed number chosen for each training iteration. Lastly, the effects of employing the α value, implementing the E2H learning strategy, and fine-tuning the DC model are discussed and analyzed.

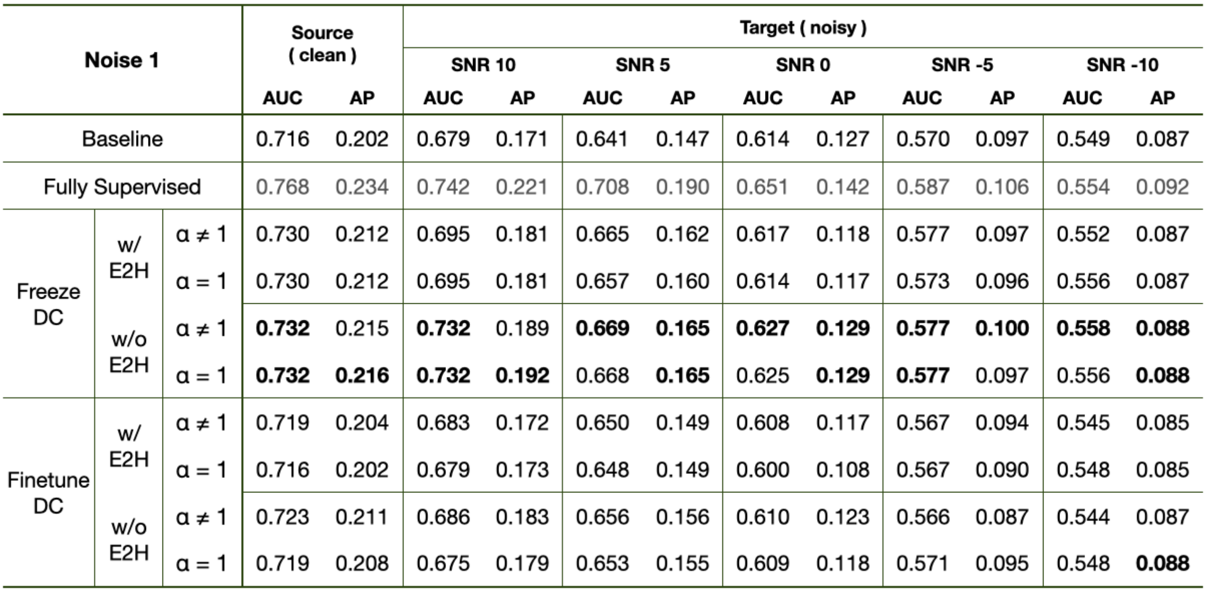
The evaluation metrics employed in this study were the AUC and AP, which were utilized to assess the accuracy and error of the multi-label classification task. Since there were six different configurations for validation and evaluation, a total of twelve values were generated for each experimental setting. These metrics provided a comprehensive assessment of the performance of the models across various validation and evaluation scenarios.



		Source (clean)		Target (noisy)									
				SNR 10		SNR 5		SNR 0		SNR -5		SNR -10	
		AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Baseline		0.716	0.202	0.679	0.171	0.641	0.147	0.614	0.127	0.570	0.097	0.549	0.089
Fully Supervised	noise 1	0.768	0.234	0.742	0.221	0.708	0.190	0.651	0.142	0.587	0.106	0.554	0.092
	noise 2	0.659	0.234	0.742	0.220	0.703	0.189	0.645	0.145	0.581	0.104	0.561	0.081
	noise 4	0.769	0.241	0.744	0.221	0.708	0.190	0.648	0.141	0.582	0.104	0.555	0.088

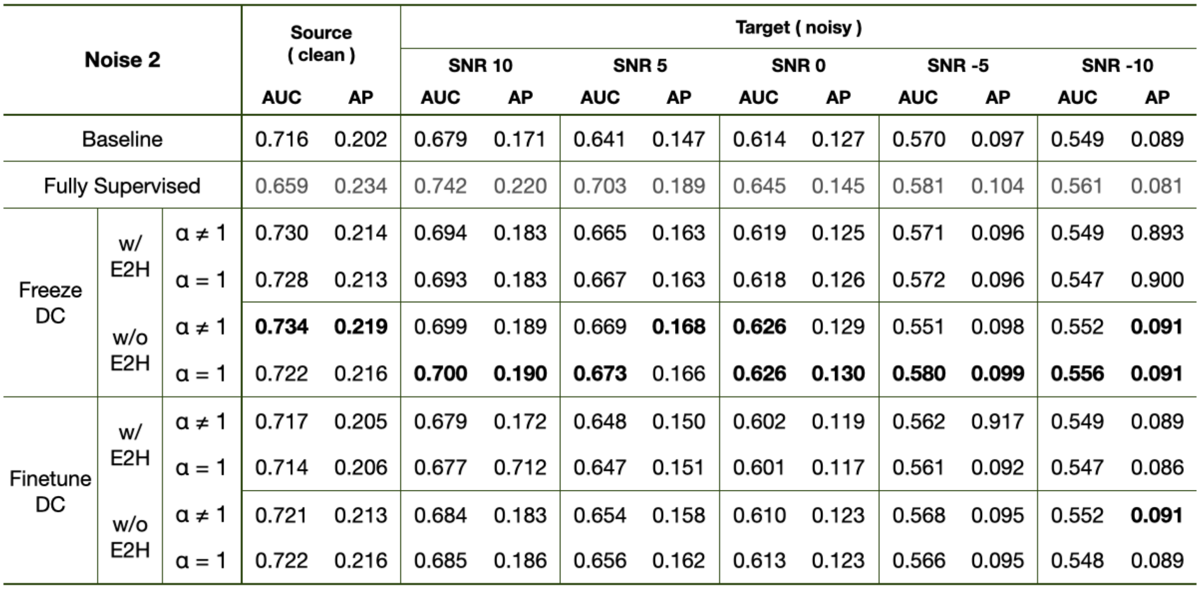
Table 4.2: Validation AUC and AP of the source(clean) and the target(noisy) domain, using the validation subset of MTAT and Audioset for noisy data.

The validation results, as depicted in Table 4.3, 4.4, and 4.5, indicate that the proposed method, incorporating an additional DC component, generally outperformed the baseline architecture. Across various experimental trials, it was observed that freezing the parameters of DC during the final stage of DAT, along with the utilization of the hyperparameter α, yielded the best performance.



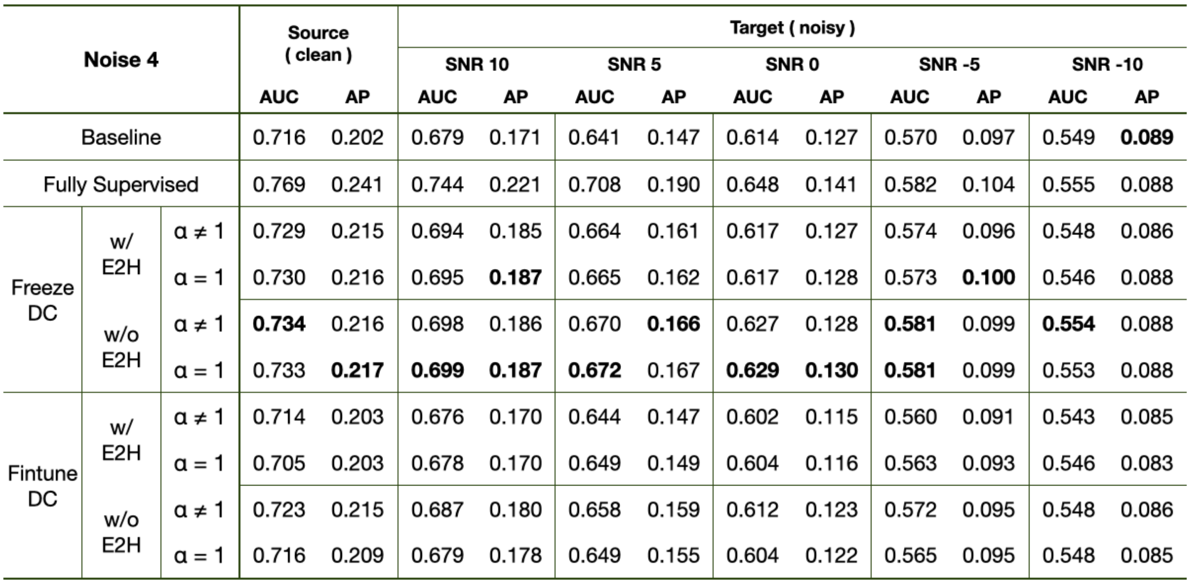
Noise 1			Source (clean)		Target (noisy)									
					SNR 10		SNR 5		SNR 0		SNR -5		SNR -10	
			AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Baseline			0.716	0.202	0.679	0.171	0.641	0.147	0.614	0.127	0.570	0.097	0.549	0.087
Fully Supervised			0.768	0.234	0.742	0.221	0.708	0.190	0.651	0.142	0.587	0.106	0.554	0.092
Freeze DC	w/ E2H	α ≠ 1	0.730	0.212	0.695	0.181	0.665	0.162	0.617	0.118	0.577	0.097	0.552	0.087
		α = 1	0.730	0.212	0.695	0.181	0.657	0.160	0.614	0.117	0.573	0.096	0.556	0.087
	w/o E2H	α ≠ 1	**0.732**	0.215	**0.732**	0.189	**0.669**	**0.165**	**0.627**	**0.129**	**0.577**	**0.100**	**0.558**	**0.088**
		α = 1	**0.732**	**0.216**	**0.732**	**0.192**	0.668	**0.165**	0.625	**0.129**	**0.577**	0.097	0.556	**0.088**
Finetune DC	w/ E2H	α ≠ 1	0.719	0.204	0.683	0.172	0.650	0.149	0.608	0.117	0.567	0.094	0.545	0.085
		α = 1	0.716	0.202	0.679	0.173	0.648	0.149	0.600	0.108	0.567	0.090	0.548	0.085
	w/o E2H	α ≠ 1	0.723	0.211	0.686	0.183	0.656	0.156	0.610	0.123	0.566	0.087	0.544	0.087
		α = 1	0.719	0.208	0.675	0.179	0.653	0.155	0.609	0.118	0.571	0.095	0.548	**0.088**

Table 4.3: The AUC and AP scores for the baseline, fully supervised, and proposed architectures with the inclusion of 1 noise in the mixture of noisy music data.



Noise 2			Source (clean)		Target (noisy)									
					SNR 10		SNR 5		SNR 0		SNR -5		SNR -10	
			AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Baseline			0.716	0.202	0.679	0.171	0.641	0.147	0.614	0.127	0.570	0.097	0.549	0.089
Fully Supervised			0.659	0.234	0.742	0.220	0.703	0.189	0.645	0.145	0.581	0.104	0.561	0.081
Freeze DC	w/ E2H	α ≠ 1	0.730	0.214	0.694	0.183	0.665	0.163	0.619	0.125	0.571	0.096	0.549	0.893
		α = 1	0.728	0.213	0.693	0.183	0.667	0.163	0.618	0.126	0.572	0.096	0.547	0.900
	w/o E2H	α ≠ 1	**0.734**	**0.219**	0.699	0.189	0.669	**0.168**	**0.626**	0.129	0.551	0.098	0.552	**0.091**
		α = 1	0.722	0.216	**0.700**	**0.190**	**0.673**	0.166	**0.626**	**0.130**	**0.580**	**0.099**	**0.556**	**0.091**
Finetune DC	w/ E2H	α ≠ 1	0.717	0.205	0.679	0.172	0.648	0.150	0.602	0.119	0.562	0.917	0.549	0.089
		α = 1	0.714	0.206	0.677	0.712	0.647	0.151	0.601	0.117	0.561	0.092	0.547	0.086
	w/o E2H	α ≠ 1	0.721	0.213	0.684	0.183	0.654	0.158	0.610	0.123	0.568	0.095	0.552	**0.091**
		α = 1	0.722	0.216	0.685	0.186	0.656	0.162	0.613	0.123	0.566	0.095	0.548	0.089

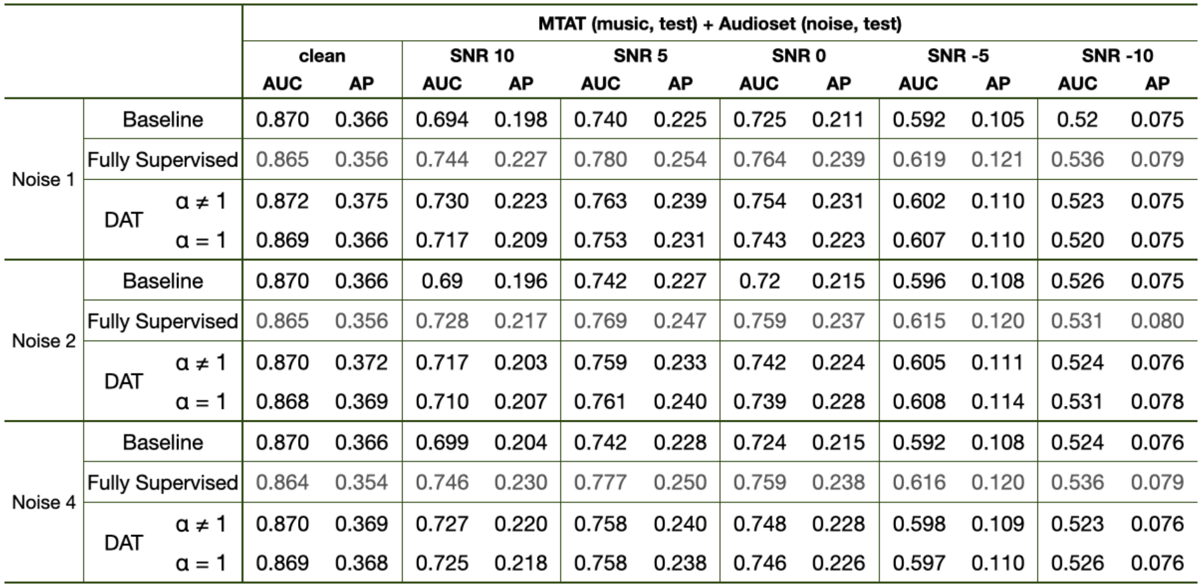
Table 4.4: The AUC and AP scores for the baseline, fully supervised, and proposed architectures with the inclusion of 2 noise in the mixture of noisy music data.



Noise 4			Source (clean)		Target (noisy)									
					SNR 10		SNR 5		SNR 0		SNR -5		SNR -10	
			AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Baseline			0.716	0.202	0.679	0.171	0.641	0.147	0.614	0.127	0.570	0.097	0.549	**0.089**
Fully Supervised			0.769	0.241	0.744	0.221	0.708	0.190	0.648	0.141	0.582	0.104	0.555	0.088
Freeze DC	w/ E2H	α ≠ 1	0.729	0.215	0.694	0.185	0.664	0.161	0.617	0.127	0.574	0.096	0.548	0.086
		α = 1	0.730	0.216	0.695	**0.187**	0.665	0.162	0.617	0.128	0.573	**0.100**	0.546	0.088
	w/o E2H	α ≠ 1	**0.734**	0.216	0.698	0.186	0.670	**0.166**	0.627	0.128	**0.581**	0.099	**0.554**	0.088
		α = 1	0.733	**0.217**	**0.699**	**0.187**	**0.672**	0.167	**0.629**	**0.130**	**0.581**	0.099	0.553	0.088
Fintune DC	w/ E2H	α ≠ 1	0.714	0.203	0.676	0.170	0.644	0.147	0.602	0.115	0.560	0.091	0.543	0.085
		α = 1	0.705	0.203	0.678	0.170	0.649	0.149	0.604	0.116	0.563	0.093	0.546	0.083
	w/o E2H	α ≠ 1	0.723	0.215	0.687	0.180	0.658	0.159	0.612	0.123	0.572	0.095	0.548	0.086
		α = 1	0.716	0.209	0.679	0.178	0.649	0.155	0.604	0.122	0.565	0.095	0.548	0.085

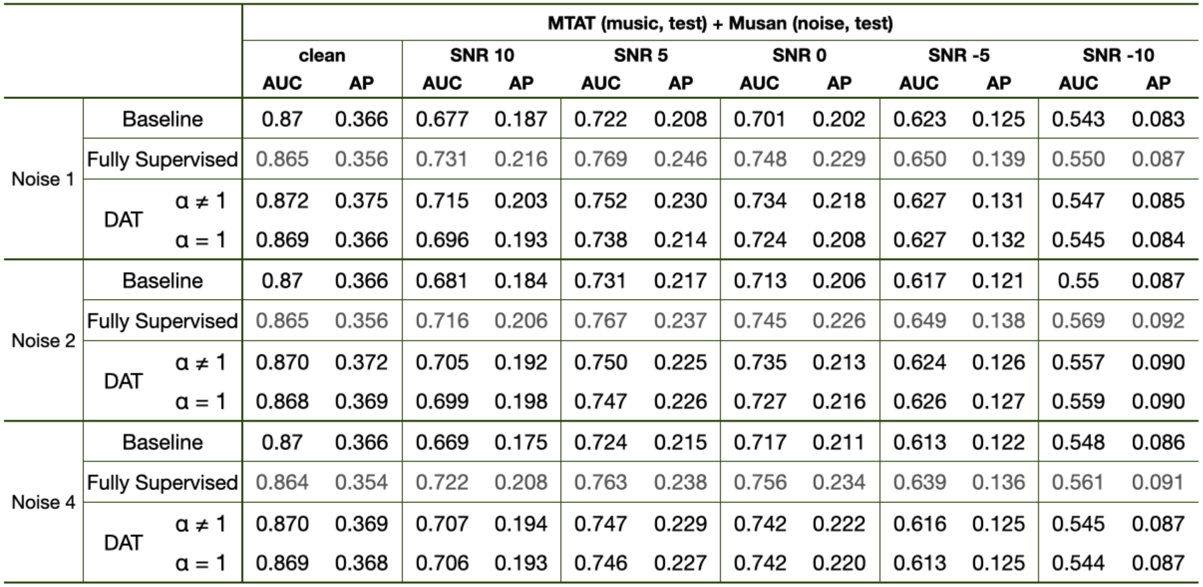
Table 4.5: The AUC and AP scores for the baseline, fully supervised, and proposed architectures with the inclusion of 4 noise in the mixture of noisy music data.

In addition to the validation results, we evaluated the best-performing models using an internal test dataset and an external test dataset. The internal dataset consisted of a combination of the test subsets of Audioset and MTAT, while the external dataset comprised all noise subsets in the Musan dataset that the model had not been exposed to before. The results, as shown in Table 4.7 and 4.8, clearly indicate that the proposed model consistently outperformed the baseline. This highlights the superior generalization capabilities of the proposed model, particularly in the presence of stronger degradations. Although the proposed method did not reach the performance level of the fully supervised model architecture, it approached it closely as the degradation became more pronounced. These findings demonstrate the effectiveness of the proposed approach in enhancing music auto-tagging accuracy, particularly in challenging noisy environments.



			MTAT (music, test) + Audioset (noise, test)											
			clean		SNR 10		SNR 5		SNR 0		SNR -5		SNR -10	
			AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Noise 1	Baseline		0.870	0.366	0.694	0.198	0.740	0.225	0.725	0.211	0.592	0.105	0.52	0.075
	Fully Supervised		0.865	0.356	0.744	0.227	0.780	0.254	0.764	0.239	0.619	0.121	0.536	0.079
	DAT	α ≠ 1	0.872	0.375	0.730	0.223	0.763	0.239	0.754	0.231	0.602	0.110	0.523	0.075
		α = 1	0.869	0.366	0.717	0.209	0.753	0.231	0.743	0.223	0.607	0.110	0.520	0.075
Noise 2	Baseline		0.870	0.366	0.69	0.196	0.742	0.227	0.72	0.215	0.596	0.108	0.526	0.075
	Fully Supervised		0.865	0.356	0.728	0.217	0.769	0.247	0.759	0.237	0.615	0.120	0.531	0.080
	DAT	α ≠ 1	0.870	0.372	0.717	0.203	0.759	0.233	0.742	0.224	0.605	0.111	0.524	0.076
		α = 1	0.868	0.369	0.710	0.207	0.761	0.240	0.739	0.228	0.608	0.114	0.531	0.078
Noise 4	Baseline		0.870	0.366	0.699	0.204	0.742	0.228	0.724	0.215	0.592	0.108	0.524	0.076
	Fully Supervised		0.864	0.354	0.746	0.230	0.777	0.250	0.759	0.238	0.616	0.120	0.536	0.079
	DAT	α ≠ 1	0.870	0.369	0.727	0.220	0.758	0.240	0.748	0.228	0.598	0.109	0.523	0.076
		α = 1	0.869	0.368	0.725	0.218	0.758	0.238	0.746	0.226	0.597	0.110	0.526	0.076

Figure 4.7: Test result of internal noise dataset, in which noisy music data is synthesized with the test subset of MTAT(music) and the test subset of Audioset(noise).



			MTAT (music, test) + Musan (noise, test)											
			clean		SNR 10		SNR 5		SNR 0		SNR -5		SNR -10	
			AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Noise 1	Baseline		0.87	0.366	0.677	0.187	0.722	0.208	0.701	0.202	0.623	0.125	0.543	0.083
	Fully Supervised		0.865	0.356	0.731	0.216	0.769	0.246	0.748	0.229	0.650	0.139	0.550	0.087
	DAT	α ≠ 1	0.872	0.375	0.715	0.203	0.752	0.230	0.734	0.218	0.627	0.131	0.547	0.085
		α = 1	0.869	0.366	0.696	0.193	0.738	0.214	0.724	0.208	0.627	0.132	0.545	0.084
Noise 2	Baseline		0.87	0.366	0.681	0.184	0.731	0.217	0.713	0.206	0.617	0.121	0.55	0.087
	Fully Supervised		0.865	0.356	0.716	0.206	0.767	0.237	0.745	0.226	0.649	0.138	0.569	0.092
	DAT	α ≠ 1	0.870	0.372	0.705	0.192	0.750	0.225	0.735	0.213	0.624	0.126	0.557	0.090
		α = 1	0.868	0.369	0.699	0.198	0.747	0.226	0.727	0.216	0.626	0.127	0.559	0.090
Noise 4	Baseline		0.87	0.366	0.669	0.175	0.724	0.215	0.717	0.211	0.613	0.122	0.548	0.086
	Fully Supervised		0.864	0.354	0.722	0.208	0.763	0.238	0.756	0.234	0.639	0.136	0.561	0.091
	DAT	α ≠ 1	0.870	0.369	0.707	0.194	0.747	0.229	0.742	0.222	0.616	0.125	0.545	0.087
		α = 1	0.869	0.368	0.706	0.193	0.746	0.227	0.742	0.220	0.613	0.125	0.544	0.087

Figure 4.8: Test result of external noise dataset, in which noisy music data is synthesized with the test subset of MTAT(music) and the test subset of Musan dataset(noise).

Chapter 5

Conclusion

5.1 Overview

In this thesis, we addressed the problem of music auto-tagging in the presence of noise, aiming to improve the accuracy of the tagging process under challenging acoustic conditions. We proposed a method of robust speech representation improving techniques that require domain adversarial training (DAT) along with the appropriate dataset. The proposed method demonstrated promising results, showcasing its potential in effectively handling noisy music data and enhancing the robustness of the tagging system.

In the initial stages of the experiment, we pre-trained the feature extractor (FE) using a large-scale music dataset, MTG-Jamendo, and a noise dataset, Audioset, and used CLMR as our backbone model. We then pretrained the domain classifier by using the embedding of pretrained FE. Lastly, we finetuned the FE using the MagnaTagATune (MTAT) dataset, incorporating both clean source domain data and noisy target domain data. Our experiments involved various configurations, including the number of noises mixed, learning strategies, and hyperparameter tuning. Through comprehensive evaluations and comparisons, we observed that the proposed model consistently outperformed the baseline architecture, demonstrating its superior generalization capabilities.

The results from the validation and test phases provided valuable insights into the effectiveness of the proposed method. During the validation phase, the proposed model achieved higher AUC and AP values compared to the baseline architecture, highlighting its ability to handle noisy music data and improve multi-label auto-tagging classification accuracy. Moreover, the performance of the proposed model approached the level of the fully supervised model architecture, particularly when dealing with stronger degradations in the target domain. This finding suggests that the proposed approach effectively generalizes to noisy music sources, even without explicit labels for the noisy target domain.

In the test phase, we further evaluated the proposed model using an internal test dataset and an external test dataset. The results consistently demonstrated the superiority of the proposed method over the baseline, both in terms of internal and external test datasets. These findings substantiate the efficacy of the proposed approach in improving music auto-tagging accuracy in the presence of noise, showcasing its potential for real-world applications.

5.2 Future Work and Limitation

In addition to the aforementioned conclusions, there are further avenues for future work and improvements in the field of music auto-tagging in the presence of noise. One potential direction is to explore the collection of real-world music-related video content data, where music tags are available. In this context, ”available” refers to instances where the same music appears in an existing dataset used for auto-tagging, rather than having explicit tags associated with the video itself.

By collecting such real-world data, researchers can enrich the training process and improve the model's performance. Incorporating this additional data would provide an opportunity to train the model on a more diverse range of music samples and enhance its ability to handle variations in musical styles, genres, and acoustic conditions. Furthermore, the inclusion of real-world data would help bridge the gap between the model's performance in controlled experimental settings and its effectiveness in real-world scenarios.

Bibliography

[1] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[2] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[3] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[4] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.

[5] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[6] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *arXiv preprint arXiv:1909.06654*, 2019.

[7] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging.," in *ISMIR*, pp. 387–392, Citeseer, 2009.

[8] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," *arXiv preprint arXiv:1606.00298*, 2016.

[9] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," 2011.

[10] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *Advances in neural information processing systems*, vol. 32, 2019.

[13] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," *arXiv preprint arXiv:2107.05677*, 2021.

[14] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *arXiv preprint arXiv:2103.09410*, 2021.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[16] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, p. 150, 2018.

[17] N. Kandpal, O. Nieto, and Z. Jin, "Music enhancement via image translation and vocoding," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3124–3128, IEEE, 2022.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[19] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[20] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fr\'echet audio distance: A metric for evaluating music enhancement algorithms," *arXiv preprint arXiv:1812.08466*, 2018.

[21] N. Schaffer, B. Cogan, E. Manilow, M. Morrison, P. Seetharaman, and B. Pardo, "Music separation enhancement with generative modeling," *arXiv preprint arXiv:2208.12387*, 2022.

[22] C. Yunkee and L. Kyogu, "Exploiting tf-conformers for general music enhancement," 2022.

[23] G. Yu, Y. Wang, C. Zheng, H. Wang, and Q. Zhang, "Cyclegan-based non-parallel speech enhancement with an adaptive attention-in-attention mechanism," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 523–529, IEEE, 2021.

[24] S. Abdulatif, R. Cao, and B. Yang, "Cmgan: Conformer-based metric-gan for monaural speech enhancement," *arXiv preprint arXiv:2209.11112*, 2022.

[25] K. P. Huang, Y.-K. Fu, Y. Zhang, and H.-y. Lee, "Improving distortion robustness of self-supervised speech processing tasks with domain adaptation," *arXiv preprint arXiv:2203.16104*, 2022.

[26] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210, IEEE, 2015.

[28] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[29] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[30] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging," ICML, 2019.

[31] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780, IEEE, 2017.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

초 록

음악 자동 태깅(Music auto-tagging)은 음악 오디오에 관련 태그(tag) 또는 레이블(label)을 할당하여 음악 검색 및 추천에 중요한 역할을 한다. 그러나 음악 정보 검색(MIR) 분야의 기존 모델은 유튜브 비디오와 같은 멀티미디어 콘텐츠에서 일반적으로 발견되는 환경 소음 및 음성과 같은 실제 소음에 직면할 때 성능 저하를 마주한다.

본 연구에서는 강건한 음성 표현 학습 방법 중점을 둔 이전 연구에서 영감을 얻어, 소음(noise)이 많은 소스에서 음악 자동 태깅의 성능을 향상시키기 위한 새로운 접근 방식을 제안한다. 우리의 방법은 도메인 적대적 훈련(domain adversarial training, DAT)을 사용하여 소음의 존재에 탄력적인, 강건한 음악 표현을 학습할 수 있도록 한다. 일반적으로 특징 추출기(feature extractor)에 대한 사전 훈련 단계에 이어 DAT 단계를 포함하는 이전의 음성 기반 연구와 달리, 우리의 접근 방식은 도메인 분류기를 위해 특별히 설계된 추가적인 사전 훈련 단계를 포함한다. 이 학습 단계를 통해 도메인 분류기는 깨끗한 음악 소스와 시끄러운 음악 소스를 효과적으로 구별하여 특징 추출기의 깨끗한 음악과 시끄러운 음악을 구별하지 않는 능력을 향상시킨다.

또한, 우리는 다양한 신호 대 잡음 비(signal-to-noise ratio, SNR)로 소음이 많은 음악 소스 데이터를 생성하는 개념을 소개한다. 모델을 다양한 수준의 소음에 노출시킴으로써 다양한 환경 조건에서 더 나은 일반화(generalization)를 촉진한다. 이를

통해 모델은 광범위한 실제 상황과 소리에 적응하고, 강력한 음악 자동 태깅을 수행할 수 있다.

우리가 제안한 구조는 음악 자동 태깅 작업에서 탁월한 성능을 보여주며, 훈련 단계에서 마주치지 않은 소음 유형에 대해서도 강건한 음악 표현을 추출한다. 이는 마주하지 않았던 소음에 대해서 잘 일반화할 수 있는 모델의 능력을 강조하여, 실제 상황에서의 효과를 더욱 향상시킨다.

이 연구를 통해 기존 음악 자동 태깅 모델의 한계를 해결하고 소음이 있는 상황에서 성능을 크게 향상시키는 새로운 접근 방식을 제시한다. 본 연구의 결과는 음악 정보 검색 분야의 발전에 기여하여 다양한 산업에서 보다 정확하고 신뢰할 수 있는 음악 분류 및 구성을 가능하게 한다.

주요어: 강건한 음악 표현, 음악 자동 태깅, 도메인 적대적 훈련

학 번: 2021-21802