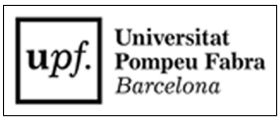
Representation Learning for Music Classification and Retrieval

: Bridging the Gap between Natural Language and Music Semantics

Minz Won

TESI DOCTORAL UPF / year 2021

THESIS SUPERVISORS
Dr. Xavier Serra i Casals and Dr. Horacio Saggion
Dept. of Information and Communication Technologies



This thesis is dedicated to my parents, Yongjin and Eunhyo.

Acknowledgements

First of all, I would like to acknowledge and give my warmest thanks to my supervisors, Dr. Xavier Serra and Dr. Horacio Saggion, who gave me the opportunity to work on this fascinating topic. I enjoyed the wonderful journey, and it was a great beginning of my research career. I would also like to thank my committee members for letting my defense be an enjoyable moment and for your valuable comments and suggestions.

I would like to express my gratitude to my mentors for giving me the chance to learn and experience cutting-edge technologies from the industry. Sanghyuk, I could successfully start my graduate study thanks to the deep knowledge that you shared. Sergio, I could conduct my first multimodal research thanks to your wise guidance. Oriol, you make Oakland great. Thank you for the onsite mentoring and supervision. Fabien, I appreciate your management, and thanks for allowing me to join the fantastic project. Justin, thanks to your passion, energy, and directing, we could make our research project successful. Nick, you always provide valuable insights when I'm lost. I could learn a lot from you. Gautham, when I thought it was impossible, you said it's possible, and you made it possible. Thank you for the supervision. Keunwoo, I learned a lot from your previous works and through collaboration. I appreciate it.

Dear MTGers, we are great. I could be a better researcher through discussions and collaborations with you. I appreciate all the moments that we shared, including seminars, workshops, conferences, Friday lunch, night outs, calçotada, etc. My thanks go out to Pablo, Jordi, Dmitry, Xavier, Edu, Frederic, Alastair, Marius, Antonio, Philip, Albin, Benno, Seva, Furkan, Juan, Lorenzo, Fabio, Andres, Pritish, and Perfe. Also, thanks to Aurelio, Lydia, Sonia, and Cristina for helping me with all the administrative work. Special thanks to the Spanish Ministry of Economy and Competitiveness linked to the Maria de Maeztu Units of Excellence Programme for the fellowship program.

I need to mention insightful external collaborators as well. Jaehun, it has been almost a decade since we started this journey at MARG. It is a great pleasure to collaborate with you, and your passion and diligence

motivate me a lot. I believe your next step will be another giant step. Jongpil and Janne, I am grateful for all the open discussions we had. You make me creative and ambitious. Cynthia, thank you for your supervision. As a young researcher, it was a great opportunity to work with you.

I would like to thank our extended family in Tirant lo Blanc, without whom I would not have been able to survive in Barcelona as an alien. Dani, Nestor, Joe, Lorenzo, Tessy, and Rousin, I will never forget all the tacos we had, music that we shared, tons of wine, sometimes silly but often serious conversations, fiestas, movie nights, pineapple pizzas, and Barça games. Special thanks to Dani, the first person who welcomed me in both Barcelona and LA. I'm curious about where is the next destination.

Many thanks to my parents, Yongjin and Eunhyo, for your genes and endless support. Although I rarely express it, I love you. My sister Songhee, I'm rooting for your bright future. You deserve it.

Last but not least, my beloved Minzita, Koni. This work would never have been possible without your support. You encouraged me to chase my dreams. You make me want to be a better man. Thank you for being an important part of my story.

Abstract

The explosion of digital music has dramatically changed our music consumption behavior. Massive digital music libraries are now available through streaming platforms. Since the amount of information available to an individual listener has increased greatly, it is nearly impossible for them to go through the entire catalog exhaustively. As a result, we need robust knowledge management systems more than ever. Recent advances in deep learning have enabled data-driven music representation learning for classification and retrieval. However, there is still a gap between machine-learned representations and the human understanding of music. This dissertation aims at reducing this *semantic gap* in order to assist listener's behavior around music information with advanced algorithmic support. To this end, we tackle three main challenges in representation learning: model architecture design, scalability, and multi-modality. Firstly, we carefully review previous deep representation models and propose new architectures that improve the representation in qualitative and quantitative ways. The newly proposed models are more flexible, interpretable, and powerful than previous ones. Secondly, training schemes beyond supervised learning are explored as a way to achieve scalable research. Transfer learning, semi-supervised learning, and self-supervised learning approaches are addressed in detail; transfer learning and semi-supervised methods are applied to enhance music representation learning. Finally, metric learning is proposed as a way to bridge music audio representation and natural language semantics, forming a multimodal embedding space. This facilitates music retrieval using arbitrary tags beyond a fixed vocabulary, and makes it possible to match music to text stories based on mood. Although our work focuses on bridging music and natural language semantics, we believe the proposed approaches generalize to other modalities. All implementation details of this thesis are available and open-source for reproducibility. The knowledge gained throughout this thesis has been put into practice and grounded in research internships and collaborations with multiple industries.

Resum

L'esclat de la música digital ha revolucionat la manera en que consumim música. Les plataformes de música per Internet posen tal quantitat d'informació i continguts a l'abast dels seus usuaris que és pràcticament impossible explorar els seus catàlegs de manera exhaustiva. Per tant, ara més que mai, cal seguir desenvolupant sistemes robustos de gestió del coneixement. Els avenços en aprenentatge profund dels darrers anys han permès el desenvolupament de mètodes per a l'aprenentatge automàtic de representacions musicals, i la seva aplicació en tasques de classificació i cerca. Tanmateix, hi ha encara un buit entre aquestes representacions apreses automàticament i la comprensió humana de la música. L'objectiu d'aquesta tesi és reduir aquest "buit semàntic", per tal d'oferir ajuda algorísmica als oients a l'hora de relacionar-se amb informació musical. A aquest efecte, abordem tres problemes de l'aprenentatge de representacions: el disseny de l'arquitectura dels models, l'escalabilitat i la multimodalitat. En primer lloc, analitzem en detall models anteriors de representació profunda, i proposem arquitectures noves que milloren les representacions qualitativa i quantitativament, donant lloc a models més potents, flexibles i interpretables. Seguidament, per tal d'assolir millor escalabilitat, investiguem processos d'entrenament més enllà de l'aprenentatge supervisat. Presentem en detall els aprenentatges per transferència, semi-supervisat i auto-supervisat; i apliquem els aprenentatges per transferència i semi-supervisat com a manera de potenciar l'aprenentatge automàtic de representacions musicals. Finalment, proposem l'aprenentatge de mètriques com a manera de reconciliar les representacions d'àudio musical i la semàntica en llenguatge natural, donant lloc a un espai d'encastament multimodal. Això facilita la recuperació de música mitjançant descriptors arbitraris en lloc de vocabularis concrets, i permet assignar música a una història automàticament en base al seu context anímic. Tot i que la nostra recerca se centra en reconciliar la música i la semàntica en llenguatge natural, opinem que el mètode proposat es pot generalitzar a altres modalitats. Tots els detalls de la implementació d'aquesta tesi estan disponibles com a codi obert per tal de permetre la seva reproducció.

El coneixement adquirit al llarg d'aquesta tesi ha estat posat en pràctica mitjançant col · laboracions amb la indústria i estades en pràctiques de recerca.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Motivation

The explosion of digital music has dramatically changed our music consumption behavior. Before digital music, listeners used to enjoy music through physical mediums such as vinyl records, cassette tapes, and compact discs. Each individual had a limited amount of libraries, so organizing the collections was manageable without special technical support. Listeners could memorize where each item was, arrange the collections in alphabetic order, make sections for different categories, or go through the collections exhaustively. Also, there are versatile human resources (sellers) at record stores who fulfill listeners' various information needs when they want to explore and discover new music. The human resources provide a well-organized music curation, help search for music, and recommend albums or artists based on listeners' tastes. Also, music catalogs provide such information to assist better music browsing.

However, nowadays, massive music libraries are available through streaming platforms, and the entire catalog is accessible by paying for monthly or annual subscriptions. The amount of information for each individual has been significantly increased. It is almost impossible to manage the entire collections item-by-item, and we interact through our mobile devices without human assistance. As a result, we need robust

	Then	Now
Medium	Physical	Online streaming
Payment	Item-based	Subscription-based
Library	Personal collections	Entire catalog
Interaction	Human resource	App interface

Table 1.1: Changes after digital music.

knowledge management systems more than ever. To this end, streaming services provide multiple functionalities that assist listeners' information behavior. They offer elaborate music curation in various ways, recommend music periodically, and afford convenient user interfaces to explore music. From academia, researchers investigate various music information retrieval techniques to support large-scale knowledge management in algorithmic ways. Table 1.1 summarizes the changes after digital music.

1.1.1 Music information behavior

How good are the current knowledge management algorithms? And how can we improve them? To answer these questions, this subsection reviews diverse music information behaviors and their relevant research topics. Especially, it mainly discusses the information behaviors that used to be at record stores through human information resources. One of the most frequent music information behaviors is *search*. We search for music using metadata such as album names, song titles, artists, and record labels. Thanks to the advance of various searching algorithms [1, 2, 3], we can search for desired songs quickly by typing the relevant metadata in the search box. Sometimes, we also use semantic tags such as genres, moods, themes, and activities to retrieve the music that we are looking for. To provide abundant high quality semantic tags, researchers have worked on metadata creation [4, 5, 6]. After the metadata creation, users can search for the labeled songs by typing semantic tags. Significantly detailed tags (e.g., Music Genome Project [4]) also enable music recommendation and

automatic playlist generation. However, manually generating semantic tags at scale is expensive as it is laborsome and requiring musical expertise. Thus, many different music classification tasks have been proposed and tackled by music information retrieval researchers to automate the labeling process. Well-known music classification tasks include genre classification [7], mood classification [8], instrument identification [9], and music tagging [10].

Another way of retrieving music is querying with audio. When we do not know any textual information (metadata), we can search music with an audio excerpt of the song or by humming the melody. Audio fingerprinting algorithms [11, 12] enable robust audio search using short audio excerpts by capturing salient audio features for music identification (i.e., audio fingerprints). Query-by-humming algorithm [13] supports users to find music with their voices by singing or humming the melody. Also, cover song identification algorithms [14, 15] help retrieving the same songs with different versions. Since machines can store more data than the number of songs humans can memorize, these algorithms sometimes surpass human abilities of query-by-audio retrieval, although cover song identification has a long way to go yet due to the task complexity.

Finally, one more important music information behavior through human resources is music recommendation. If we recall the interaction at record stores, sellers recommend music based on the customer's previous purchase history, which implies the customer's taste. Or sometimes, the recommendation is based on the acoustic similarity of given example songs or the artists. Also, they recommend music based on their knowledge. For example, when different artists are from the same label, when different albums belong to the same category, or when different songs use similar instrumentation, this prior knowledge can be helpful in music recommendation. The introduced multiple facets of music recommendation can be found in recommender systems (RecSys) research. Collaborative filtering [16, 17] is a domain-agnostic recommender system that filters items based on user-item history. By factorizing a huge user-item matrix, lower-dimensional vectors can represent user preferences and item characteristics. As music streaming platforms emerge, large-scale user-item

Information behavior	Query	Relevant research topics
Search / retrieval	Text	Searching algorithms, metadata creation, music classification
Music identification	Audio	Audio fingerprinting, query-by-humming, cover song identification
Recommendation	Text, audio, purchase history	Collaborative filtering, content-based recommendation, music similarity, knowledge graphs,

Table 1.2: Music information behaviors and their relevant research topics.

data have been accumulated. As a result, the collaborative filtering systems became extremely powerful. However, the system has two inherent issues: cold-start problem and popularity bias [17]. It cannot handle new items or new users when they do not have enough interactions (cold-start). Popular items are likely to be recommended more, which results in a filter bubble (popularity bias). To alleviate the drawbacks, content-based recommendation [18] has been explored. Instead of relying on user-item interactions, content-based approaches recommend music based on acoustic similarity [19, 20] learned by the representation models. Although content-based approaches do not suffer from the aforementioned issues of collaborative filtering, it only considers music audio among various factors that affect our music consumption and recommendation (e.g., artist relationship, background). Knowledge graphs help exploit a collection of structured data by transforming the music information into useful knowledge [21, 22, 23]. Table 1.2 summarizes all introduced music information behaviors and their relevant research topics.

The primary motivation of this dissertation is to assist music listeners when they browse and explore music through mobile interfaces without the support of human information resources. Then the question is, "are these algorithm-based applications good enough to replace human agents to fulfill our information needs?" We are already experiencing huge successes in metadata search and audio-based search. Their speed and accuracy surpass what human agents can perform, and the scalability is incomparable. Also, massive user-item interaction enabled extremely pow-

erful music recommender systems using collaborative filtering. But in the case of semantic search and content-based recommendation, still there is a gap between algorithms and humans. Machine-perceived music is yet far from human-perceived music. Only with a tiny modification in music audio (e.g., additive white noise), the machine's predictions can be dramatically changed, although the modification is very trivial for human perception [24, 25]. Then the next question is, "how can we reduce this semantic gap?"

At the core of semantic applications, there is music representation learning. Learned representation enables music classification for semantic search, similarity-based music retrieval, and content-based music recommendation. Learned representations are in a format of high-dimensional vectors. We want the vector representations to be robust and flexible like human understanding of music. Human understanding of music can be often described in our language. We can represent a song with a few words (music tags). Or, we can also depict a song in detail with natural language. To this end, this dissertation aims at *reducing the semantic gap* between machine-perceived music and human-perceived music by (*i*) using advanced music representation learning algorithms in *scalable* and *data-driven* fashions, and (*ii*) bridging the learned music audio representations with our natural language semantics to form a multimodal representation space.

1.1.2 Representation learning

Robust music representation or feature is the key to success in previously introduced music information retrieval (MIR) systems. Through decades, MIR researchers worked on the manual design of music representation. They carefully designed representations (features) based on their domain knowledge, and the extracted features are used to perform various information retrieval tasks. For example, mel-frequency cepstral coefficient (MFCC) [26] is widely used in timbre-related tasks (e.g., instrument identification) by taking advantage of domain knowledge that harmonic periodicity is crucial in timbre recognition. As another example, harmonic

pitch class profile (HPCP) [27] uses our musical domain knowledge in its feature design to enhance chord recognition models.

Although the manual design strategies are effective in machine learning, the design process is cumbersome and sometimes the representation fail to extract discriminative information for the task. Instead of relying on human ingenuity, the representation design process can be performed in a fully-data-driven fashion by learning with general priors: i.e., *representation learning* [28]. As representation is learned from the data, it is easier to extract useful (relevant) information. Especially, deep learning based approaches are rapidly growing and consistently reporting remarkable performances in many domains including computer vision [29] and natural language processing [30].

MIR researchers also have actively adopted the deep representation learning to tackle different MIR problems. Learned deep representation enabled both generative [31, 32] and discriminative models [33]. And the input sources are not only limited to audio but also include MIDI [34] and scores [35]. Among these various tasks and input sources, this dissertation focuses on deep representation learning of *music audio* to perform *discriminative* tasks since the main motivation is to support information behaviors of music listeners: music classification and retrieval. The following section discusses current limitations of deep music audio representation learning in discriminative tasks.

1.2 The problem

This section diagnoses the limitations of recent advances in music audio representation learning in three aspects: representation model, scalability, and multimodality.

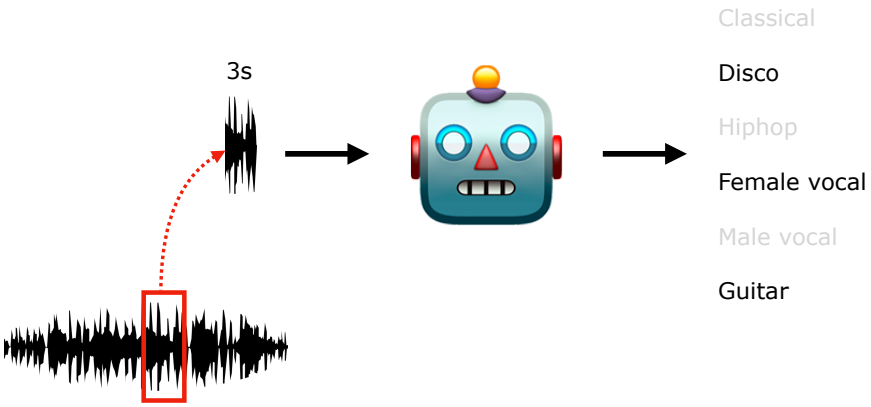


Figure 1.1: Instance-level music representation learning.

1.2.1 Representation model

Evaluation

As deep learning emerges, music representation learning research has entered a new phase, and many data-driven approaches have been proposed. However, researchers sometimes use jargon in various ways, some implementation details and evaluation methods are ambiguously described in the papers, and they use slightly different experimental setups (e.g., dataset splits, library versions, computing environments, and optimization methods). Thus, it is not easy to compare different music representation models directly with each other and there can be unintended miscommunication between researchers. It guides to an overly optimistic or overly pessimistic baseline that results in comparing experimental results that should not be compared. It creates tremendous obstacles that impedes scientific improvements in music representation learning.

Instance-level music representation

As shown in Figure 1.1, most current music representation models use short audio excerpts as their inputs (instance-level training). One song is

cropped into multiple instances, then the model needs to predict the musical attributes (tags) based on the acoustic characteristics of the instance. It is mainly because the instance-level training shows better performance and because it is not easy to increase the size of the receptive fields effectively. Instance-level representations are aggregated later to represent a song-level information using global summarization such as global max pooling, global average pooling, and majority voting. That means current music representation models behave like a bag-of-feature model [36] instead of representing music as a sequence.

There are two issues of the instance-level training. Firstly, it is different from human music perception. If we mix the order of the audio excerpts, for example, the latent representation of the model won't be changed because the model only cares about the existence of features not their sequence, while humans perceive it differently. As we would like to build a semantic space that resembles human understanding of music, this difference is not desirable.

Another issue is the multiple instance problem [37]. When a random excerpt is cropped from a song with a tag *piano*, there is a possibility that the excerpt does not include any piano sound in it, because the tag *piano* does not imply the piano appears at every time step. However, the model is trained to predict the excerpt to have a tag *piano*. In this way, there is a possibility that the model learns some other acoustic biases that exist in piano music. For instance, the model can predict any jazz music to have a tag *piano* because some audio excerpts of jazz piano music were used as positive examples of *piano* during the training, even if the excerpts do not have any piano sound in them. Details of instance-level training is further discussed in Section 2.2.

Interpretability

Deep learning models are often described as a black box. Different from manually designed features or rule-based models, it is difficult to understand the underlying mechanism of deep learning models because they are formed by the composition of multiple non-linear transformations. This

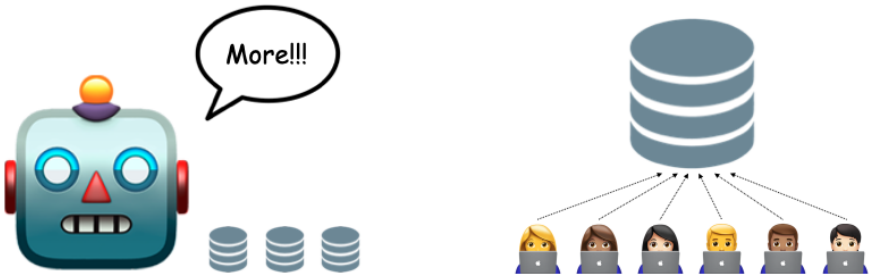


Figure 1.2: Irony of large-scale manual labeling to train a better model.

makes it difficult to diagnose, understand, and fix the model. Hence, it can be a major bottleneck for model development. To this end, there have been previous works for visualizing and understanding latent features of deep learning models. For images, it is very intuitive to understand the learned patterns by visualizing (highlighting) the relevant part. Also, for natural language processing, highlighting relevant words can already provide a lot of information to be used. However, when we apply the same approach to the audio, it is still hard to understand the learned features. Some parts of mel spectrograms will be highlighted but the visual information of audio is not intuitive for humans.

Less interpretability is also caused by the architectures that we use. Convolutional neural network (CNN) is originally designed to tackle computer vision problems not audio. We are using it because of its powerfulness in pattern recognition but the mechanism is far from the known concept of human audio recognition. CNN models an audio input as a snapshot of an image while humans perceive the audio as sequence. For better understanding of the model behaviors, audio-specific interpretability research is needed.

1.2.2 Scalability

Deep learning models are data-hungry. In general, model performance gets better with more data. As a result, ironically, we end up demanding a

large amount of human efforts again during the dataset creation process, although the original purpose of developing the models was to unburden human efforts (Figure 1.2). The inclusion of human elaboration hinders scalable representation learning due to the labeling cost. So there is a demand for different training strategies instead of labeling more data to scale up.

In a real-world scenario, we may have a large-scale music library, although only a few of them might have manual labels. With supervised learning scheme, abundant unlabeled data are discarded during the training process and only few labeled data are used. Under the limited circumstance, we need other training schemes beyond supervised learning to incorporate other labeled data and large-scale unlabeled data. In the field of computer vision and natural language processing, various training methods, such as transfer learning, semi-supervised learning, and self-supervised learning, have been explored. These approaches are generalizable in many domains including music representation learning, because their concepts are not task specific.

1.2.3 Multimodality

Various music classification models enable tag-based music retrieval. Users can search for music with genres, moods, and instruments using predicted tags by the classification models. However, still there is a gap between the music tags and human understanding of music. Pretrained music classification models are limited to fixed vocabulary hence the model cannot handle unseen tags even if they are synonyms (*happy* and *happiness*) or acronyms (*EDM* and *electronic dance music*) of the tag in the training set taxonomy. This gap can be interpreted as a semantic gap between tag-level representation and music audio representation.

As another example of the semantic gap, current music semantic space is limited to a single modality: audio. However, human forms multimodal semantic spaces. We can match appropriate music for a video clip (video-to-audio), or evoked emotions from a book can be applied to our music selection (text-to-audio). By bridging the gap between music audio se-

mantics and other modalities, music retrieval can be more flexible beyond tag-based search. Also, there is a possibility that the semantic distribution of one modality can complement to form a more robust music semantic space. Nevertheless, multimodal approaches for music representation learning is less explored yet. This dissertation explores multimodal representation learning, particularly aims at bridging the gap between *natural language* and *music audio semantics*.

1.3 The solution

The main idea of advanced music representation learning in this research is three-fold: (*i*) propose better music representation models by tackling the problems of current representation models discussed in the previous section, (*ii*) adopt transfer learning and semi-supervised approaches to step further beyond supervised learning, (*iii*) explore multimodal embedding spaces to reduce the gap between natural language and music audio semantics.

Representation model

Firstly, we implement and reproduce a variety of previous music representation models and compare their performances under the same computational environment, datasets, preprocessing, evaluation metrics, and libraries. We can assess the baselines reliably, and each model's pros and cons will be revealed through the process.

Through the holistic evaluation and analysis of music representation models, we will learn useful insights for building better architectures. Based on the acquired knowledge, we build more powerful music representation models, and they are assessed using the same evaluation pipeline. More data-driven ideas are included here, and a new sequence modeling technique (i.e., Transformer [38, 30]) is actively used to achieve long sequence modeling and interpretable systems.

Scalability

The issue of limited amount of labeled data can be tackled in various ways. We can use a large-scale labeled data from other relevant tasks (source tasks) to train a model. Then transfer the learned representation to solve our target task by fine-tuning the model. This transfer learning scheme takes advantage of the knowledge from already existing labeled data. Although we do not have genre labels for large-scale data, for example, we have artist labels for most tracks. As every artist has his/her/their own musical styles, we can train a discriminative model that predicts artist labels as a source task, then transfer the learned representation to solve downstream target tasks [39]. We explore transfer learning of artist classification to perform more robust music representation learning.

Instead of only relying on labeled data, we can also exploit unlabeled data. Semi-supervised learning is a machine learning approach that utilizes both (small-scale) labeled data and (large-scale) unlabeled data. It takes advantage of the best of two worlds: strong supervision and scalability. We introduce successful semi-supervised schemes to music representation learning to achieve the best performing music classification models.

Multimodality

Finally, we reduce the gap between natural language and music semantics by jointly learning multimodal embedding spaces. Previously explored music representation models are used to represent music semantics. And pretrained word / sentence / paragraph embeddings are used to represent text semantics. Finally, two different modalities are bridged together via multimodal metric learning to form a joint embedding space. Multiple representation strategies are discussed within each modality, and we explore various training schemes to enable multimodal representation learning.

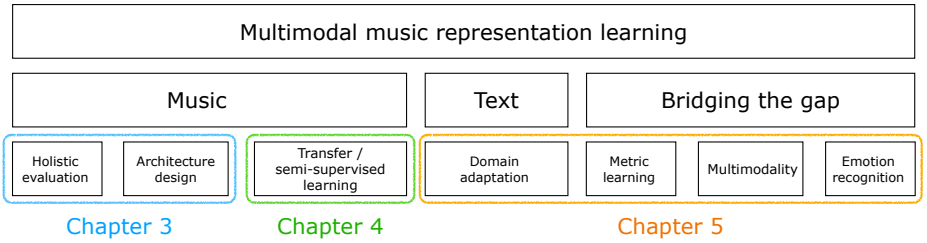


Figure 1.3: Thesis overview.

1.4 Summary of contributions

The contributions of this thesis are summarized as follow. Extensive comparison of existing representation models was performed. Based on the insights from the experiment, new state-of-the-art models are proposed for automatic music tagging. More data-driven front end is proposed to preserve harmonic characteristics, and Transformer is introduced to enable long sequence modeling with better interpretability. The representation models are further improved with various training schemes including transfer learning and semi-supervised learning.

Multimodal approaches are also explored to reduce the gap between natural language and music semantics. Domain-specific musical word embeddings are released, and an advanced open-vocabulary music retrieval system is introduced. The multimodal embedding research is extended to allow paragraph-level inputs which facilitates matching appropriate music to text based on their mood and emotion.

This dissertation aims at open science and reproducible research. All experiments are performed using public datasets and I contributed to build a new open-source dataset (MTG-Jamendo dataset [40]). All implementation details are open-sourced, and pretrained models are available online (see Appendix A).

1.5 Thesis outline

This thesis is structured as follows: Chapter 2 introduces relevant concepts and previous works of representation learning that include music classification, various training schemes, and natural language processing. Then Chapter 3 revisits previous representation models and proposes advanced architectures for music representation learning. Chapter 4 steps further beyond supervised learning using transfer learning and semi-supervised learning. It enables more scalable music representation learning with extra data. Chapter 5 presents multimodal approaches to reduce the gap between natural language and music semantics. Multimodal models facilitate flexible semantic search beyond fixed vocabulary, and also allow sentence-/paragraph-level inputs. Chapter 6 draws conclusions, summarizes contributions, and discusses future directions. Figure 1.3 summarizes the outline of the thesis.

Chapter 2

BACKGROUND

2.1 Introduction

This chapter introduces relevant concepts, topics and previous related works to achieve the goal: multimodal music representation learning for classification and retrieval. Firstly, music classification is explained in the next subsection (Section 2.2). It describes the definition, different classification tasks, input representations, evaluation metrics, instance-level training, various types of music information, and supervised learning. Music classification enables semantic search using predicted music tags (e.g., genres, moods). But also, deep-learning-based music classification is an important area of music representation learning. Learned latent representation from the class supervision can be transferred to solve other downstream tasks, and also can be utilized to form a music similarity space to recommend music. After reviewing music classification, Section 2.3 introduces training methods beyond supervised learning. Training schemes such as transfer learning, semi-supervised learning, and self-supervised learning are discussed. As scalability is a key to success in deep representation learning, these training strategies are highly important. Section 2.4 introduces recent advances in natural language processing to incorporate one more modality in music representation learning. Starting from techniques to represent word-level semantics [41, 42], it

discusses recent dramatic improvements in natural language processing which enables versatile text representation in a self-supervised manner [38, 30].

To summarize, this chapter introduces ingredients including music representation learning (Section 2.2), scalable representation learning (Section 2.3), and text representation learning (Section 2.4), which are relevant for tackling multimodal music representation learning.

2.2 Music classification

2.2.1 Overview

Music classification is a music information retrieval (MIR) task whose objective is the computational understanding of music semantics. For a given song, the classifier predicts relevant musical attributes. Based on the task definition, there are a nearly infinite number of classification tasks – from genres, moods, and instruments to broader concepts including music similarity and musical preferences. The retrieved information can be further utilized in many applications including music recommendation, curation, playlist generation, and semantic search.

In the deep learning era, the role of music classification is not only limited to semantic search but also includes music representation learning. Through the training process using class supervision, useful music representation is learned, and the representation can be transferred to solve other relevant problems, or the learned embeddings can be used to measure content-based music similarity. Since scalability matters in deep representation learning, and music classification datasets (e.g., million song dataset [43]) are the most scalable labeled datasets in MIR, improving music classification models is crucial to music representation learning. In this subsection, we review relevant concepts of music classification research.

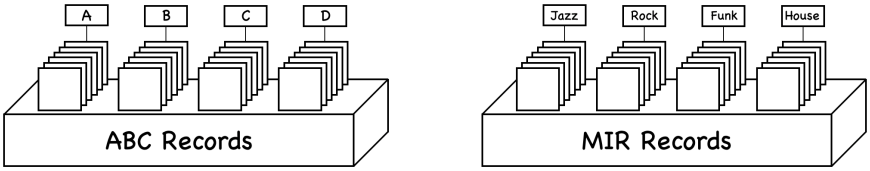


Figure 2.1: Two record stores with different knowledge management strategies.

Single-label classification

Let's say there are two record stores in the town (Figure 2.1). 『ABC Records』 curates all the records in alphabetic order, while 『MIR Records』 categorizes their stocks based on musical genres. When a customer already knows what he/she wants to buy, 『ABC Records』 is a good place to go as he/she can search for the item by the alphabetic index. However, when a customer wants to browse and discover new music, 『MIR Records』 will be preferable as he/she can visit the section with favorite musical genre. Like this, well-designed categorization (i.e., music classification) helps customers browse music more efficiently. This record store scenario can be interpreted as a single-label classification task. One item can be in a single section; hence categories (genres in this example) are exclusive[1].

Multi-label classification

Different from the record store example, one item may belong to multiple categories. For example, one song can be *disco* and *K-pop* simultaneously, and these categories are not exclusive to each other. Also, listeners would like to browse music by instruments, languages, moods, or context, not only musical genres. We can handle these multiple musical attributes with multi-label classification. The multi-label classification is often referred to as "music tagging" since it puts various music tags for a given song.

[1]Genres are not always exclusive to each other. One song can belong to multiple genres.

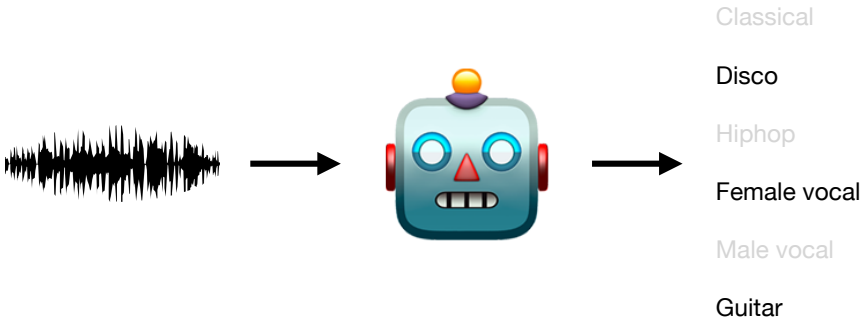


Figure 2.2: Multi-label music classification.

As shown in Figure 2.2, multi-label classification is handled as a binary classification for each musical attribute. For each label, the system determines whether a given song is positive to the label or not. In contrast with single-label classification, labels are not exclusive, and multiple tags can exist together.

2.2.2 Music classification tasks

There can be an almost infinite number of music classification tasks based on product requirements. Among them, the most explored music classification tasks in MIR research are: genre classification [7], mood classification [8], instrument identification [9], and music tagging [10]. Music tagging subsumes all other classification tasks as any class (label) can be musical tags.

Genre classification

Music genre is one of the most prevalent categories to describe music. When talking about musical preferences, many people assume they are supposed to talk about their favorite genres. The simplest problem formulation of genre classification is to define a genre taxonomy that is flat and mutually exclusive (single-label classification). This is how the pioneer-

ing Gtzan genre classification dataset was constructed [7]. It comprises ten high-level genres: *blues, classical, country, disco, hip hop, jazz, metal, pop, reggae,* and *rock*. With the idea of mutual exclusiveness of music genres, more datasets have been proposed including Ballroom dataset [44], FMA-small and FMA-medium [45], and ISMIR 2004 genre [46].

The different problem formulation appears in more modern datasets. The mutual exclusiveness assumption was loosened in the million song dataset [43] (with tagtraum genre annotations [47]). This allows a track to have more than one genre label (i.e., multi-label classification). Finally, a hierarchical genre taxonomy is considered in datasets such as FMA-Full [45] and AcousticBrainz-Genre [48].

Mood classification

Although the genre boundaries are already unclear, mood is more subjective by its definition. Also, there is the difference between perceived mood (the mood of music) and induced mood (the mood one would feel when listening to the music). Mood classification also includes both single-label (e.g., MoodsMIREX) and multi-label datasets [43, 40]. But due to its unclear boundaries, researchers tried to formalize the task as a regression problem by allowing continuity in mood annotations. They labeled mood annotation in a two-dimensional plane where the axes represents arousal and valence [49, 50]. Sometimes one more dimension is added (dominance) to form a three-dimensional space, or time-varying mood annotations are provided [51].

Instrument classification

Instrument identification is more objective compared to genre and mood classes. In early stage, instrument identification dataset was also built assuming mutual exclusiveness. The IRMAS dataset [52] has a single predominant instrument tag for each item. More recently, instrument identification is treated as multi-label classification in dataset such as OpenMIC-2018 [53].

Tackling instrument identification incorporates multiple instance learning. Acoustic characteristics of certain genres or moods may span over the entire sequence. But in case of instruments, a song will be labeled with an instrument tag even if it appears temporarily. Multiple instance learning is depicted in Section 2.2.5.

Music tagging

The progress of the computer and internet has given the privilege of labeling music to every single music listener – the democratization of annotation. Social music services gathered these tags, and predicting the collected tags from the audio content became a task named automatic music tagging.

There is no constraint on which tag to be labeled, hence music tags are diverse and noisy. In the million song dataset (MSD) [43], for example, there are 505,216 tracks with at least one music tag. And the number of unique tags is 552,366. There are more unique tags than the number of tracks. If we take a closer look at their distribution, some music tags are extremely subjective. The 7th popular tag is *favorite*, the 18th is *Awesome*, and the 33rd is *seen live*. The 37th is *Favorite* and the 41st is *Favourite*.

However, still there are many tags that are relevant to the music content. After removing the ambiguous tags, the top-15 tags include *rock, pop, alternative, indie, electronic, female vocalists, dance, 00s, alternative rock, jazz, beautiful, metal, chillout, male vocalists,* and *classic rock*. There are genre, mood, and instruments, each of which has been treated as a target category for music classification. Like this, music tagging subsumes all other music classification tasks.

Although music tags are noisy, it is easier to collect them than collecting (expert-annotated) genre, mood, or instrument labels. As a result, scalable datasets are available in music tagging, which enables deep learning approaches in music classification. This is one reason why most previous music representation models are explored with music tagging datasets such as MagnaTagATune (MTAT) [54] and MSD [43].

2.2.3 Input representations

In traditional music tagging approaches [55], carefully designed features, such as mel-frequency cepstral coefficients (MFCC), zero-crossing rate (ZCR), and chromagrams, have been used as inputs of music tagging models. However, modern deep learning approaches only do minimal feature engineering or directly utilize the raw data as inputs so that the model can learn useful representations from data. This subsection introduces common input representations for data-driven music classification.

Raw audio

One core idea of deep representation learning is to learn useful representation from data instead of manual feature design. Relevant features are learned from the training data distribution. Some previous works [56, 57, 58] tackled music classification in an end-to-end fashion by using raw audio waveform as their inputs. Although it shows comparable results in music classification, their performances are yet lower than other approaches using small feature engineering (e.g., short-time Fourier transform). Since the model is assumption-free, it is more flexible but the search space to learn is enormous compared to preprocessed inputs. There is a possibility that the raw audio model to outperform feature-based approaches when a larger-scale dataset is available. But at current scale, small feature engineering is still required to achieve the best results.

Mel spectrogram

Mel spectrogram is one of the most common input representation of music classification models. It resembles the known physiology of human auditory system: tonotopy. In human cochlea, there is the basilar membrane. The thickness and width changes at each region of the basilar membrane, hence each region vibrates at certain frequency. The tonotopic frequencies are logarithmically distributed (Figure 2.3).

The first step of processing mel spectrogram is short-time Fourier transform (STFT). The output of the STFT is called spectrogram. Spec-

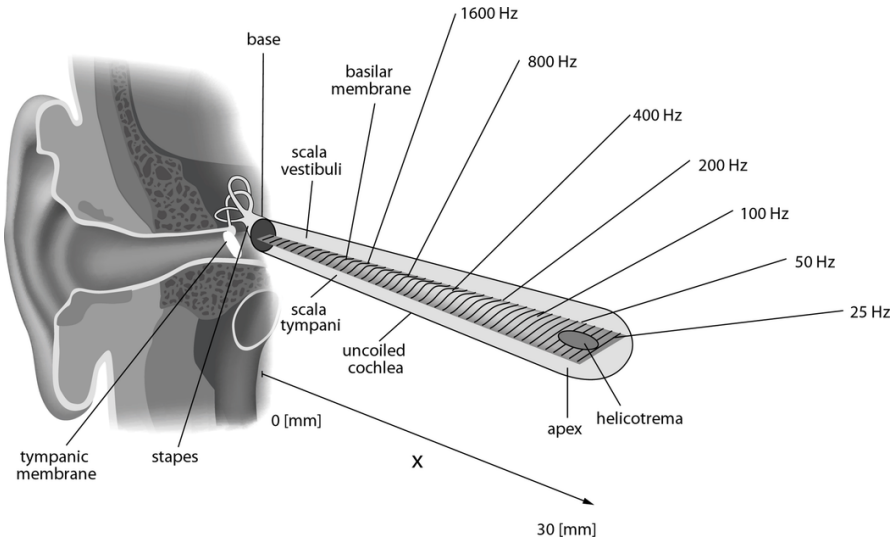


Figure 2.3: Tonotopy in the basilar membrane of the cochlea. Image from Wikipedia.

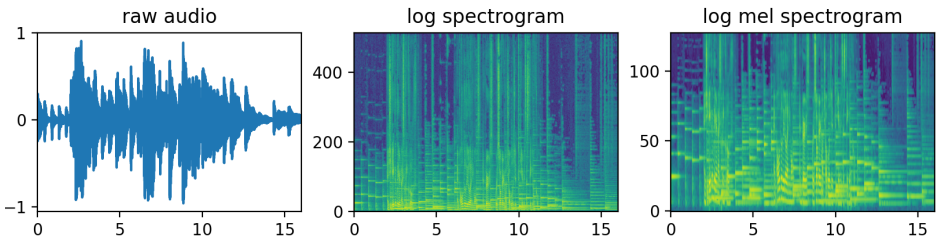


Figure 2.4: Different input representations for music classification.

trogram is a two-dimensional representation where x-axis is time, and y-axis is frequency (Figure 2.4 - middle). Each frequency bin includes the phase and magnitude of certain frequency (tonotopy) at each time step. Then a logarithmically distributed triangular filters (i.e., mel filterbank) are applied to the frequency axis of the spectrogram. As a result, we get the mel spectrogram (Figure 2.4 - right). In most cases, phase information is discarded and only magnitude is used. Finally, log scale is used for the magnitude to form a log-magnitude mel spectrogram. The whole steps are formalized as follow:

$$S = 10 \cdot log(|mel(STFT(x))|) \tag{2.1}$$

where x is raw audio waveform, $STFT$ is short-time Fourier transform, mel is a mel filterbank, and S is log-magnitude mel spectrogram.

As shown in Figure 2.4, spectrogram is easier to understand than raw audio as it provides time-varying energy at each frequency band. Mel spectrogram looks similar to spectrogram but it is more compact, and the low frequency region has allocated more bins than it does in spectrogram.

2.2.4 Evaluation

Evaluation of models is one of the most crucial parts of music classification. No matter how many state-of-the-art models are available, the practical performance of the application can be different depending on which model we choose. Hence, proper evaluation metrics fit for purpose are essential in the model selection. This subsection explores widely used evaluation metrics of music classification with a simple example case.

Figure 2.5 shows an example of a binary classification task. We want to assess a classification model that detects vocals in music. The dataset has ten songs with vocal (blue dots) and ten songs without vocal (orange cross marks). The green circle is a decision boundary of the model. The model predicts that the items in the green circle are vocal music, and the items at the outside of the circle are instrumental music.

As shown in Figure 2.6, the predictions can be separated into four categories.

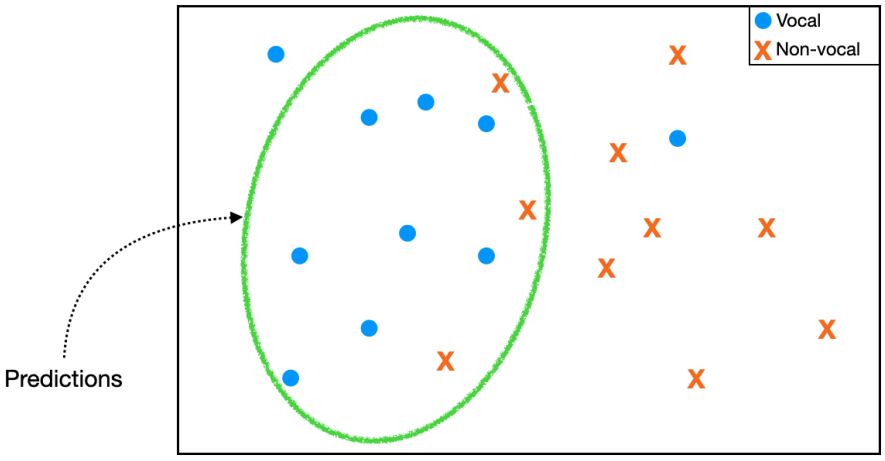


Figure 2.5: An example of single-label binary classification.

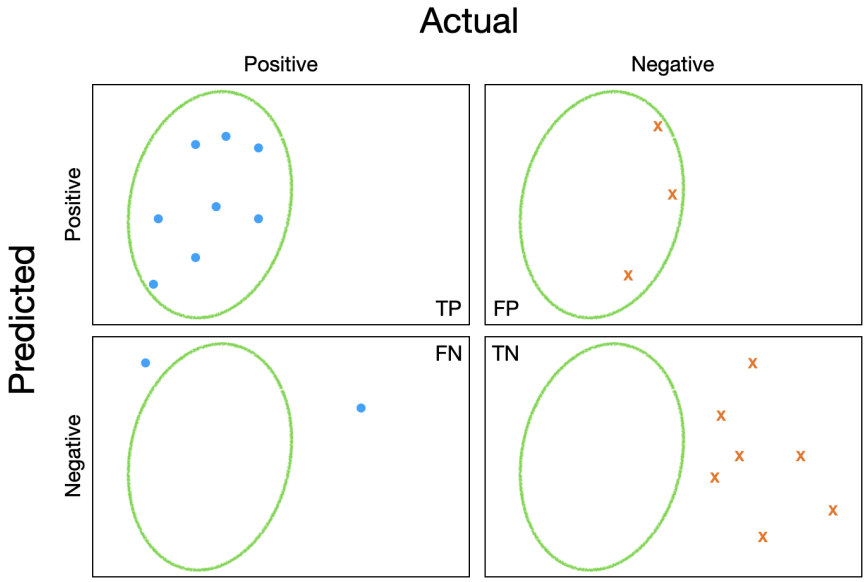


Figure 2.6: Four categories of predictions: true positives (upper left), false positives (upper right), false negatives (lower left), and true negatives (lower right)

- True positives (TP): Correctly predicted vocal music.

- False positives (FP): Predicted as vocal music but they are non-vocal music.

- False negatives (FN): Predicted as non-vocal music but they are vocal music.

- True negatives (TN): Correctly predicted non-vocal music.

Accuracy

Accuracy is an intuitive and most widely used evaluation metric to assess classification models. It measures how many items are correctly classified. The formula of accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

In the given example, accuracy is 0.75. More intuitively, 15 items among 20 are correctly classified no matter they are vocal or non-vocal music.

Precision

Precision measures how many retrieved items are truly relevant. In retrieval systems, precision is important as it indicates how accurate the retrieval results are. The model retrieved 11 items in the green circle to be relevant (i.e., vocal music). Among them, 8 songs are truly vocal music, and 3 songs are not. The formula of precision is:

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

hence precision is 0.7273 in our example.

Recall

Recall measures how many relevant items are correctly retrieved. Among 10 positive songs with vocal, 8 of them are correctly predicted as vocal music while 2 of them are rejected (two blue dots at the outside of the green circle). The formula of recall is:

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

hence recall is 0.8 in this example. Recall is also known as sensitivity or true positive rate. And the opposite term is specificity or true negative rate: how many of negative items are correctly rejected, i.e., TN / (FP + TN).

F-measure

High precision is important in building a reliable retrieval system because users can trust the system when retrieved items are truly relevant. However, a high precision / low recall system only retrieves a few positive items, which end up with low diversity. In other words, we can achieve high precision by applying a very strict threshold (narrower decision boundary), but a lot of relevant items will be discarded (false negatives) due to the high threshold.

F-measure or F-score considers both precision and recall. The traditional F-measure (F1-score) is defined as the harmonic mean of precision and recall. The maximum value is 1.0, and the lowest is 0 (either precision or recall is zero).

$$F_1 = \frac{precision \cdot recall}{precision + recall} \tag{2.5}$$

Depending on system requirements, either precision or recall may be more critical. In that case, the balance of precision and recall can be parameterized in Fbeta-measure. Fbeta-measure has one more coefficient β that controls the weights between precision and recall.

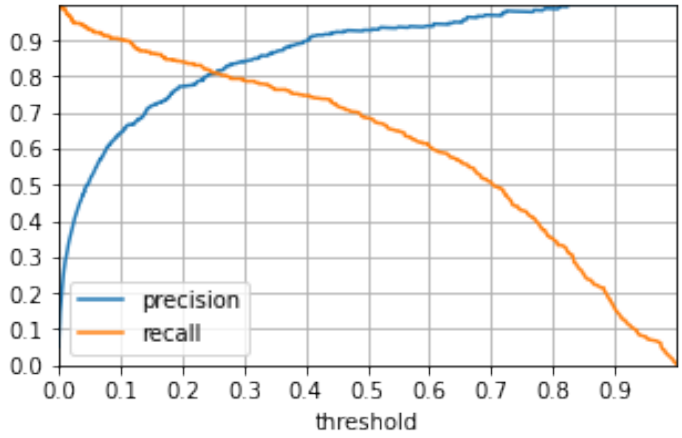


Figure 2.7: Threshold-varying precision-recall curve.

High precision vs high recall

The classification model outputs the likelihood of the input to have vocal between 0 and 1. Hence, to make a final decision, we need to set a threshold. With a high threshold, the model becomes more strict, which means the green circle becomes narrower in Figure 2.5. As a result, the retrieved results by the model for a given query "vocal music" will be reliable. However, the model only retrieves a few songs among the entire vocal tracks (i.e., high precision and low recall). This can be observed from the precision-recall curve in Figure 2.7. As the threshold gets closer to 1.0, precision goes higher while recall goes lower. On the other hand, if the threshold gets lower, it results in high recall and low precision, which means the system returns any item to be positive. Like this, appropriate decision making of threshold is crucial in classification tasks.

Area under receiver operating characteristic curve (ROC-AUC)

The receiver operating characteristic curve (ROC curve) reflects the model's threshold-varying characteristics. The ROC curve is created by plotting true positive rate (TPR) against false positive rate (FPR), where TPR is

Figure 2.8: Receiver operating characteristic surve.

also known as sensitivity or recall, and FPR is calculated as (1 - specificity). Specificity is also known as true negative rate (TNR).

In Figure 2.8, a dotted black line indicates the ROC curve of a random classifier, a blue line indicates a better classifier, and an orange line shows a perfect classifier. As a classifier gets better, the area under the curve (AUC) gets wider. The area under the ROC curve is referred to as the ROC-AUC score. The maximum value is 1.0, and the lowest is 0.5 (a random classifier).

Area under precision-recall curve (PR-AUC)

It is known that ROC-AUC may report overly optimistic results with imbalanced data [59]. Therefore, the area under the precision-recall curve (PR-AUC) is often provided together with ROC-AUC. The precision-recall curve is created by plotting precision against recall at different thresholds. Unlike the ROC-AUC score, which has 0.5 as its lowest value, the lowest bound of PR-AUC differs by data. When a model predicts every item to be positive regardless of threshold, the recall will always be

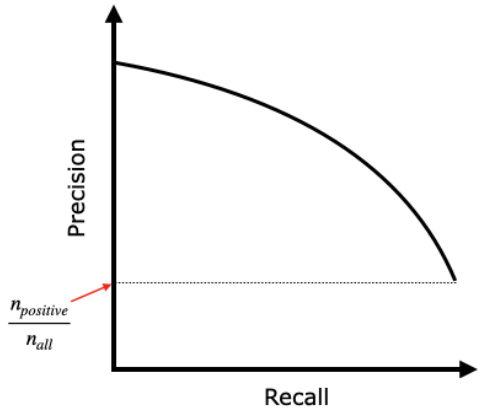


Figure 2.9: Precision-recall curve.

1.0, and precision will be a ratio of positive items w.r.t. all items. Hence, the lowest value of PR-AUC is the ratio of positive items. It penalizes when the retrieval performance is low for the less represented tags. There are multiple ways of calculating PR-AUC, and the average precision [2] is one method for calculating PR-AUC. There are other methods such as trapezoid estimates and the interpolated estimates. This dissertation uses the average precision.

2.2.5 Multiple instance learning

Music signals are in the form of sequential data. In this sequence, regarding typical tags, some acoustic characteristics may appear locally (e.g., instruments) while some others may span over the sequence (e.g., mood, genre). For example, when a song has a tag *female vocal*, it does not imply that the female vocal appears in every time segment of the song. However, a *cheerful* mood can be perceived from the long sequence. This means a successful music classification model needs to be able to extract both local and global features.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html

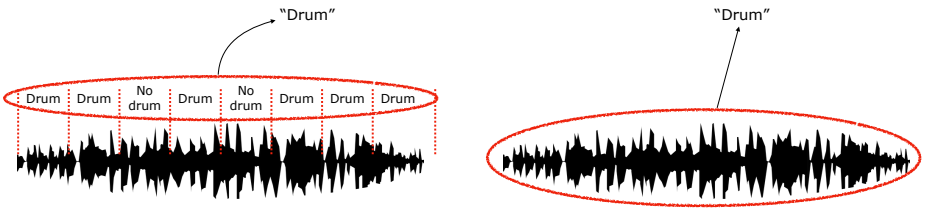


Figure 2.10: Instance-level (left) vs sequence-level inference (right).

In essence, the former case (i.e., detecting local characteristics from a sequence) is a multiple instance problem [37]. A given music signal (a bag of multiple instances) will be labeled with a tag if a part (an instance) of the signal has a certain relevant acoustic characteristic. In most cases, we do not have time precise instance-level annotations because the precise labeling can be laborsome, therefore a music tag associated with a song is simply applied to all music excerpts (instances) of the song during training [60].

Furthermore, in many music classification models, they do not input the entire sequence of the song. Short audio excerpts (3 to 5 second-long) are cropped from the entire sequence, then input to the model. This instance-level training is justified by our intuition – humans can predict music tags within just a few seconds. For example, people would not spend 3 minutes to determine whether a track is *rock*. However, there is a possibility that the instance receives a wrong supervision since the relevant acoustic characteristic does not appear in the instance. After the instance-level training, during the evaluation phase, the instance-level predictions are aggregated with a method such as majority voting, global max pooling, global average pooling, or adaptive pooling [60]. Most music tagging models use this approach due to the following reasons. Firstly, as multiple chunks (instances) can be collected from the entire song, there can be more training examples. Secondly, the task gets more difficult since the model needs to learn from short audio excerpts. Thirdly, large-batch training is available with short audio because they can fit into limited memory. Finally, it shows better performance than using longer inputs.

Although most music classification models use instance-level training and inference, the MIL problem can be also tackled in an end-to-end fashion (Figure 2.10). The end-to-end networks model the entire sequence, and capture both local and global acoustic characteristics in it. However, this approach results in less training samples compared to the aforementioned instance-level training hence less generalizable when the same amount of songs are used for training.

2.2.6 Three types of music information

In a previous work for knowledge management of musical metadata [61], the authors proposed to categorize musical metadata into three categories (i.e., *editorial*, *cultural*, and *acoustic*) based on the nature of the labeling process. In the original work, the term "music metadata" and "music information" are used interchangeably, but in this dissertation, I use the term "music information" instead of "metadata" to incorporate broader concepts beyond music tags.

Editorial information is literally collected from editors. Album names, artist names, song titles, record labels, and decades are included. As most tracks already have editorial information, we don't need any further step to collect labels. Some semantic tags such as genres and moods can be also editorial information if they are provided by the editors. But in this research, I do not include semantic tags as editorial information because the original source of the semantic tags come from cultural or acoustic aspects of music. Note that, in this dissertation, editorial information only includes "objective" written information of the album and the artist.

Cultural information is produced by environments and culture. For some genres, such as *K-pop* and *detroit house*, although they have certain acoustic characteristics, the cultural background of the song is critical to determine the genre. Even if a song sounds similar to K-pop music, if the song is performed by non-Korean artists, mainly consumed by non-Korean listeners, and written in non-Korean lyrics, it is not a K-pop song (although we can say the song sounds like K-pop). Like this, cultural and environmental factors are important music information. A widely used

way of collecting cultural information is collaborative filtering [62]. From the user-item interaction history, user information and item information can be decomposed by matrix factorization. In modern streaming platforms, the collaborative filtering using large-scale user-item matrix provides abundant cultural information.

Finally, acoustic information is collected from the audio contents. From low-level features, such as MFCC, beat, and tempo, to high-level semantics, such as genres and moods, various musical aspects are included in acoustic information. Most music classification tasks assume that musical categories or tags can be predicted using acoustic information.

Although there are three categories, their boundaries are sometimes unclear. Also, they have a strong correlation. An artist (editorial) may have a strong fan base in certain country (cultural). And the artist may have a signiture sound or style of music (acoustic). Some training schemes introduced in the next section takes advantage of these correlations to overcome limited amount of labeled data.

2.2.7 Supervised learning for music classification

This subsection introduces a step-by-step process of supervised learning for music classification — see Table 2.1. Firstly, given raw audio x is preprocessed (line 3). In this step, if we want to train a model with instance-level training (Section 2.2.5), a short audio excerpt is randomly cropped from the sequence. Then the input is processed into other representation (e.g., mel spectrogram) as we reviewed in Section 2.2.3, or raw audio can be directly passed so that the model can learn useful representation from data. The processed feature is input to the representation model $\mathcal{M}$ (line 4). The model can use any deep representation learning architectures including convolutional neural networks (CNN), recurrent neural networks (RNN), or Transformer. The deep representation r passes through a projection layer $\mathcal{N}$ to make a final prediction (line 5). Multi-layer perceptrons (MLP), also known as a fully-connected layer, is used for projection. Finally, a training loss between prediction p and ground truth label y is calculated (line 6), then the model ($\mathcal{M}$ and $\mathcal{N}$) is updated using gradient

Supervised Learning

Input audio X, labels Y
Modules preprocessing $\mathcal{P}$, representation model $\mathcal{M}$, projection layer $\mathcal{N}$
Functions loss function $\mathcal{L}$, back propagation $\mathcal{B}$
Train

```
1  for x ∈ X, y ∈ Y
2    do
3      s ⟵ P(x)              // preprocessing
4      r ⟵ M(s)              // get representation
5      p ⟵ N(r)              // prediction
6      l ⟵ L(p, y)           // get loss
7      M, N ⟵ B(M, N, l)     // model update
6    end do
7  end for
```

Table 2.1: Pseudocode of supervised learning.

descent (line 7).

When the output classes are exclusive to each other (i.e., single-label classification), softmax activation function is used in the projection layer $\mathcal{N}$ so that the sum of the scores to be 1.0.

$$f_{softmax}(x)_i = \frac{e^{x_i}}{\sum_k e^{x_k}} \tag{2.6}$$

On the other hand, in multi-label classification tasks, sigmoid activation function is used. The output scores are in a range between 0 and 1.

$$f_{sigmoid}(x_i) = \frac{1}{1 + e^{-x_i}} \tag{2.7}$$

Note that, if there are only two classes in single-label classification, softmax (Equation 2.6) and sigmoid (Equation 2.7) are identical. After passing through the activation function, the model is trained to minimize a cross entropy loss,

$$CE = -\sum y_i \log \hat{y}_i \tag{2.8}$$

where y_i is ground truth and $\hat{y}_i$ is prediction score. Since mean squared error (MSE) loss does not penalize misclassified items enough, cross entropy loss with softmax / sigmoid makes model converge faster in classification tasks, hence more frequently used. MSE loss is more widely used in regression problems.

In evaluation phase, instead of cropping one random excerpt in preprocessing, it returns multiple short audio excerpts from the sequence. Then the predictions of each excerpt in line 5 are aggregated to make a final prediction. Global max pooling, global average pooling, or majority voting are typically used for the aggregation. Finally, the model performance is assessed using various evaluation metrics introduced in Section 2.2.4.

2.3 Beyond supervised learning

Although supervised deep learning approaches report outstanding performance in many music classification tasks, collecting scalable data remains a challenge. Data acquisition in music is especially challenging due to following reasons. Firstly, it takes time to listen to the music. In case of images, human agents can label images in few seconds. We can answer whether an image is a dog or a cat, immediately. Also, in case of speech recognition or sound event detection (e.g., dog barking, vacuum cleaner), the acoustic events occur in a very short time period. It only takes few seconds to label the utterance or the sound event. However, in case of music, we need to listen to the music for three to five minutes to fully understand the content correctly. For example, we can determine the genre or mood without listening to the entire sequence, but we need to check every time step to determine the existence of the instruments. Secondly, strong domain knowledge is required to label musical attributes. Distinguishing musical genres, subgenres, and styles are difficult without musical knowledge. And instrument labeling is not easy without previous experience and training. Finally, a lot of music semantics are not objective. Genres do not have clear boundaries, the boundaries may differ by

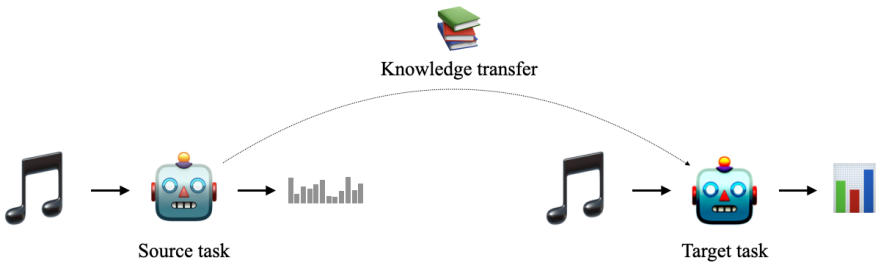


Figure 2.11: Transfer learning.

cultural background, and moods are extremely subjective. That means we need multiple human agents to label on the same item so that the labels to have certain agreements as ground truth.

Due to the aforementioned challenges in data labeling, there is a demand for other solutions for scaling up music representation learning without labeling more data. In a real-world scenario, there are large-scale music libraries, but only a few of them might have manual labels. Also, sometimes, there is a discrepancy between the taxonomies of the existing training data and the target task. In the circumstances, one can take advantage of existing labeled data to solve other relevant problems, use a pretrained model to give a supervision to a student model, inject consistency constrains so that the model to return noise-agnostic results, regularize the entropy of the model, or ensemble aforementioned various approaches. This section introduces representation learning methods beyond supervised learning that utilizes external labeled data (transfer learning) or unlabeled data (semi- and self-supervised learning).

2.3.1 Transfer learning

The core idea of transfer learning is to (*i*) learn knowledge from solving a problem (source task) and (*ii*) apply the knowledge to solve other relevant problems (target task) [63] — see Figure 2.11. For example, if the model is able to perform instrument identification (source task), the learned knowledge would be useful to solve music genre classification

Transfer Learning

Input source data X, source labels Y, target data X', target labels Y'
Modules preprocessing $\mathcal{P}$, representation model $\mathcal{M}$,
source projection layer $\mathcal{N}$, target projection layer $\mathcal{N}'$
Functions loss function $\mathcal{L}$, back propagation $\mathcal{B}$
Train with source task
for $x \in X$, $y \in Y$
do
$s \longleftarrow \mathcal{P}(x)$ // preprocessing
$r \longleftarrow \mathcal{M}(s)$ // get representation
$p \longleftarrow \mathcal{N}(r)$ // prediction
$l \longleftarrow \mathcal{L}(p, y)$ // get loss
$\mathcal{M}, \mathcal{N} \longleftarrow \mathcal{B}(\mathcal{M}, \mathcal{N}, l)$ // model update
end do
end for
Train with target task
for $x' \in X'$, $y' \in Y'$
do
$s \longleftarrow \mathcal{P}(x')$ // preprocessing
$r \longleftarrow \mathcal{M}(s)$ // get representation
$p \longleftarrow \mathcal{N}'(r)$ // prediction
$l \longleftarrow \mathcal{L}(p, y')$ // get loss
$\mathcal{M}, \mathcal{N}' \longleftarrow \mathcal{B}(\mathcal{M}, \mathcal{N}', l)$ // model update
end do
end for

Table 2.2: Pseudocode of transfer learning.

(target task) since as the underlying concepts in music genres are related to instrumentation. The assumption is that although the source and target tasks are not identical, if the dataset for the source task is much larger than the target task data, transferring the learned knowledge could lead to a better performance.

Table 2.2 depicts the transfer learning flow. The first half (line 1-7) is identical to supervised learning (Table 2.1). A model is pretrained using source task data X and the labels Y. Afterwards, the model is optimized to solve the target task using X' and Y'. But in this step, a pretrained model $\mathcal{M}$ from the source task is used instead of training a model from scratch. Both representation model and the projection layer can be updated during the training, or they can be partially updated. Sometimes, only a part of the model is used so that latent features in the middle of the network to be transferred for prediction. In a previous work [64], the authors pretrained a music tagging model with the million song dataset (MSD). Then the model was transferred to solve downstream tasks such as genre classification, emotion recognition, audio event classification, etc. In the experiments, a pretrained model outperformed the baseline MFCC features and random-weights CNN features in six different tasks.

The introduced transfer learning experiment takes advantage of music tags in the MSD which are mostly acoustic information (e.g., genre, instrument). However, collecting those music tags still requires human effort of labeling, and some tags are subjective. Different from acoustic information, editorial information is objective and easier to collect. For example, artist labels are already provided with most songs. Artist labels are more objective, songs from the same artist tend to share prominent musical characteristics, and it is easy to scale up without manual labeling. Park et el. [39] transferred learned representation from artist labels to solve music genre classification, and reported performance gain in three different datasets. From the similar motivation, we submitted a challenge winning submission [65] to the Recognize Music Genre from Audio challenge [66] in *The Web Conference* 2018. We transferred learned representation from artist classification to tackle genre classification. In this process, instead of targeting thousands of artists directly, we clustered artists into fewer classes: artist group factors (AGFs). This avoids possible bottlenecks caused by large number of classes, and prevents data sparsity. Details of the AGFs are introduced in Section 4.2.

Another way of scaling up music representation learning in a transfer learning scheme is to use cultural information. When users use mu-

sic streaming services, their listening history is cached in database. As a result, abundant user-item interaction data is available without manual labeling efforts. Large-scale data is accumulated as time passes. This huge user-item matrix are factorized into lower dimensional vectors [67] to represent user embeddings and item embeddings using collaborative filtering [16, 17]. Since user's musical taste is affected by acoustic characteristics, one can take advantage of the latent factors to supervise deep representation models. A previous work [18] trained deep convolutional neural networks to predict latent factors from music audio (i.e., item embeddings). The pretrained model facilitated music similarity space for content-based music recommendation, and the data distribution showed that similar genres to be grouped together in the learned representation space.

2.3.2 Semi-supervised learning

In many realistic scenarios, we have limited labeled data and abundant unlabeled data. For example, in the million song dataset (MSD) [43], only 24% of them are labeled with at least one of the top-50 music tags. As a consequence, most the existing MSD tagging research discarded the 76% of the audio included in MSD. One can label the 76% to improve the performance, but there is another way of incorporating the large-scale data. Semi-supervised learning is a machine learning approach that utilizes both (small-scale) labeled data and (large-scale) unlabeled data. In general, semi-supervised models are optimized to minimize two loss functions: a supervised loss, and an unsupervised loss:

$$Loss = Loss_{supervised} + \lambda \cdot Loss_{unsupervised} \tag{2.9}$$

where the ratio between two loss functions is controlled by a hyper parameter λ. Semi-supervised learning is a broad concept of a hybrid approach of supervised learning and unsupervised learning. There are many variants of designing the unsupervised loss.

Self-training

Input labeled data X, labels Y, unlabeled data Z
Models teacher model $\mathcal{T}$, student model $\mathcal{S}$
Functions loss function $\mathcal{L}$, data augmentation $\mathcal{A}$,
back propagation $\mathcal{B}$

Train

1 **for** $x \in X, y \in Y$
2 **do**
3 $p \longleftarrow \mathcal{T}(x)$ // predict
4 $l \longleftarrow \mathcal{L}(p, y)$ // get loss
5 $\mathcal{T} \longleftarrow \mathcal{B}(\mathcal{T}, l)$ // update teacher model
6 **end do**
7 **end for**
8 **for** $x \in X, y \in Y, z \in Z$
9 **do**
10 $p_1 \longleftarrow \mathcal{S}(x)$ // predict
11 $l_1 \longleftarrow \mathcal{L}(p_1, y)$ // get supervised loss
12 $\psi \longleftarrow \mathcal{T}(z)$ // generate pseudo-label
13 $p_2 \longleftarrow \mathcal{S}(z)$ // predict
14 $l_2 \longleftarrow \mathcal{L}(p_2, \psi)$ // get semi-supervised loss
15 $\mathcal{S} \longleftarrow \mathcal{B}(\mathcal{S}, l_1 + l_2)$ // update student model
16 **end do**
17 **end for**

Table 2.3: Pseudocode of self-training.

Self-training

Most self-training [68] approaches follow the teacher-student pipeline. Table 2.3 depicts a step-by-step process of the self-training. Firstly, a teacher model is trained with labeled data in a supervised fashion (line 1-7). Then a student model is optimized to predict the labels of labeled data (line 10-11), and the pseudo-labels of unlabeled data (line 13-14).

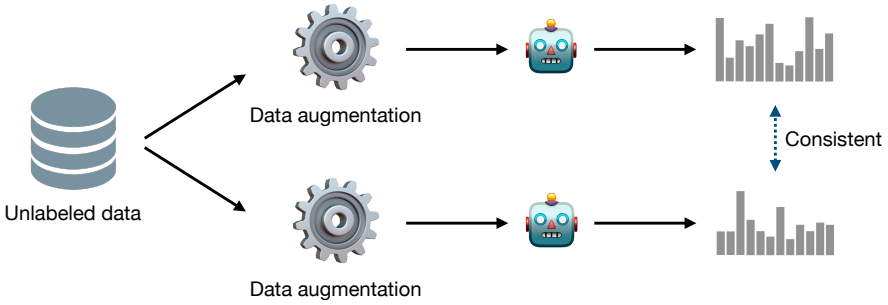


Figure 2.12: Consistency training.

Pseudo-labels are generated by predicting the labels of unlabeled data using a pretrained teacher model (line 12). If the task is a multi-label classification, the output of the teacher model will be a score between 0 to 1 since sigmoid activation function is used in the output layer. These scores can be used as pseudo-labels as they are (soft-label), or binarized with certain thresholds so that the outputs to be hard-labels.

Consistency training

Consistency training [69] constrains models to generate noise invariant predictions. When there is an apple on the table, for example, it is always an apple even if we take a look at it from a different angle or a different distance, under different lights, or through glass. The view changes but the original property "apple" does not change. As another example, when we listen to jazz music, no matter which speaker we use, whether people are speaking, or the audio is time-stretched, it is still jazz music. Like this, consistency training injects various noise that do not harm the original property, and optimizes the model to return consistent predictions. Unsupervised loss of consistency training is formalized as follow:

$$Loss_{unsupervised} = D(p(y|A(x), \theta), p(y|A(x), \theta)) \tag{2.10}$$

where x is an unlabeled input, A is stochastic data augmentation, and D is a distance metric such as mean squared errors. Note that the data

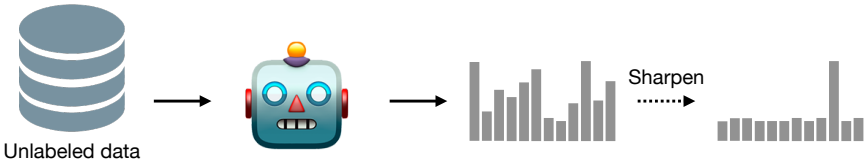


Figure 2.13: Entropy regularization.

augmentation A is stochastic; hence the two outputs in Figure 2.12 are different.

Entropy regularization

Entropy regularization minimizes the entropy of the model's predictions (Figure 2.13). A straightforward implementation is to directly minimize the entropy of the predictions for unlabeled data [70]. Along with the supervised loss, the model makes predictions of unlabeled data. Although we do not know the ground truth labels, the model needs to minimize the entropy of the predictions. This entropy regularization can be also achieved in an implicit manner by training with one-hot encoded pseudo-labels [71]. In this case, the model first makes a prediction using unlabeled data. The prediction is then modified to be an one-hot encoded vector and used as a pseudo-label to supervise the model. In both approaches, the training scheme incorporates unlabeled data by adding minimum entropy regularization in explicit [70] or implicit [71] ways which prevent the model from making ambiguous decisions.

More approaches

There are more semi-supervised learning methods such as graph-based approaches [72] and generative modeling [73]. Or introduced multiple semi-supervised approaches can be combined together. MixMatch [74] incorporates entropy minimization, consistency regularization, and MixUp [75]. Noisy student training [76] is another successful semi-supervised

learning scheme that takes advantage of self-training and consistency regularization. Noisy student training will be discussed in detail in Chapter 4.

2.3.3 Self-supervised learning

Self-supervised learning or unsupervised learning learns useful representation from data without labels. It is easy to scale up with self-supervised learning because it does not require any labeling process. Pretrained representation is further optimized with downstream datasets in a transfer learning scheme. Self-supervised learning is gaining more attention than ever as scalability emerges as one of the most important factors in deep representation learning.

Contrastive learning

We previously reviewed consistency regularization in semi-supervised learning. This consistency regularization can be applied to self-supervised learning. Contrastive learning is a method that learns representations by modeling similarity from natural variations of data. In Figure 2.12, a single item is encoded into two different "views". When there exist N training examples, there are $2N$ views. If we choose one view, there is only one positive pair among $2N-1$ views. Contrastive learning optimizes the model to minimize the distance between positive pairs and maximize the distance between negative pairs.

In momentum contrast (MoCo) [77], a dictionary of examples in the data is maintained as a queue. Each example in the mini-batch is encoded, and put in front of the queue, while the last item in the dictionary is subsequently dequeued. The pretext task used in MoCO is to define a contrastive loss on the query and the keys of the dictionary: a query matches the key when the query is an embedding of a different view of the same data point. For example, if the query is an embedding of the bassoon solo in Stravinsky's *Rite of Spring*, it should match with the key that corresponds to the *Rite of Spring*. The encoded query should be similar to its corresponding key and dissimilar to other keys in the dictionary.

Training a MoCo encoder is done with positive and negative pairs of examples in a mini-batch. The positive example pairs are made of queries that correspond to keys of the current mini-batch. The negative pairs are queries of the current mini-batch and keys from past mini-batches. The keys are encoded by a "slowly progressing" encoder because the dictionary's keys are drawn over multiple mini-batches. This encoder is implemented as a momentum-based moving average. Therefore there are two encoders: an encoder for the queries and a momentum-encoder for the keys. The main difference between these two encoders is how they are updated. The query encoder is updated by backpropagation, while the momentum encoder is updated by linear interpolation of the query and the momentum encoder.

SimCLR [78] is a simple contrastive learning approach to learn robust visual representation. It leverages strong image data augmentations, large batch sizes, a single large encoder, and a simple contrastive loss to pretrain an encoder. For each image example in the mini-batch, two augmented views are taken. This is done by a series of data augmentations that are applied randomly to each example. Each of these augmented views are embedded using a standard ResNet [79] encoder. When learned representation is used in downstream tasks, the embeddings are projected to a different latent space by a small linear layer on which the contrastive loss is computed. This idea has been experimented in music representation learning [80] and showed comparable results in music classification.

Autoregressive models

Auto-regressive models are optimized to predict next time steps in the sequence based on previously observed information. When we listen to music, listen to a speech, read a sentence, or look at an image, there are expected next steps that preserve the pattern's continuity.

Contrastive predictive coding (CPC) [81] is a universal framework of representation learning. The data can be an image in which neighboring patches usually share spatial information locally. In the case of speech signals, it could be the phonemes that should be similar to the neighbors.

In music, the chorus of a song is expected to repeat in another part of our audio signal. These related observations are mapped similarly as a representation in a latent space in CPC. The main hypothesis is that predictions of related observations are often conditionally dependent on similar, high-level pieces of latent information. Firstly, complex natural data, such as images and audio, are compressed into a latent embedding space. Then an autoregressive model uses the latent representation to make predictions for future observations. These observations are mapped to the corresponding representation. In this process, a contrastive loss is used instead of directly predicting future values. The idea is further improved to learn more powerful speech representation [82, 83].

Jukebox [32] is an autoregressive model that learns music audio representation. Jukebox comprises two parts: compressing music to discrete codes and learning autoregressive distribution. The first part uses a vector-quantized variational autoencoder (VQ-VAE) [84] to compress audio to a discrete space. Then the encoded music codes are used to optimize an autoregressive transformer [38]. The transformer is a powerful sequence model that uses a self-attention mechanism. The transformer is trained to predict future music codes, and the model can perform as a music generator. But the learned representation is not limited to generative tasks, and it also showed versatility in many music classification tasks [85].

Other approaches

Problem-agnostic speech encoder (PASE) [86] learns speech representation with multi-task learning. Each task aims at predicting the features that we already know. The model needs to reconstruct the original waveform (autoencoder), predict log power spectrum, mel-frequency cepstral coefficients (MFCC), etc. Also, sequence predictive coding is included. By solving multiple tasks, the model can learn problem-agnostic speech representation.

More recently, transformer variants showed huge successes in self-supervised representation learning. Bidirectional encoder representations from transformer (BERT) [30] initially proved its versatility in natural

language processing. BERT is optimized in two different ways. Firstly, a part of the sentence is masked, and the model is trained to predict the missing tokens. Secondly, two different sentences are concatenated, and the model needs to predict whether they are originally continuing sentences or two irrelevant sentences. This idea has been expanded towards image processing [87], video processing [88], and more recently, speech recognition [89]. Section 2.4.3 describes the transformer in detail.

2.4 Natural language processing

2.4.1 Why NLP?

This section introduces recent advances in natural language processing (NLP). Before that, why do we need to adopt NLP in music representation learning? As introduced in Section 2.2, deep learning approaches report huge successes in music classification. The predicted musical attributes are further used in music retrieval and recommendation. However, still, there is a semantic gap between machine-learned representation and human understanding of music. Machines are not flexible enough to handle synonyms (e.g., *happy* and *happiness*) and acronyms (R&B and Rhythm Blues). Machines often fail to generalize when they encounter unseen types of data. Different from machines, human music perception is multimodal. We take advantage of other modalities when we listen to music. Not only audio but also visual cues and written information contribute to our music understanding. In this dissertation, I'm aiming at (*i*) learning multimodal music representation that can handle flexible music tags beyond fixed vocabulary and (*ii*) reducing the semantic gap by taking advantage of information from text modality. The following subsections review pretrained word embeddings and transformer architectures to achieve the goals.

2.4.2 Word embedding

In most classification tasks, each label is represented as a one-hot encoded vector. These vectors do not connote any word semantics since they are simply indices in a vocabulary. On the contrary, Word2Vec [41] represents words as vectors that have multiple degrees of similarity in the embedding space. In the pretrained embedding space, simple algebraic operations are available using the word vectors. For example, *vector*("King") − *vector*("Man") + *vector*("Woman") results in a *vector*("Queen").

There are two ways of training the Word2Vec embeddings: continuous bag-of-words (CBOW) and continuous skip-gram. To represent a word, CBOW gets neighboring words as its inputs. Inputs are embedded as vectors, and the vectors are averaged. Finally, the averaged vector is projected to represent the center word. For example, from the sentence "I drink water every morning", we average the embedding vectors of "I", "drink", "every", and "morning", then project their average to represent the word "water". In training data, other words, such as *coffee* and *milk*, may appear in an analogous context; hence they are semantically similar in the trained embedding space. Continuous skip-gram is similar to CBOW, but the direction is opposite. It uses the current word as an input to predict the neighboring words.

Different from CBOW and continuous skip-gram, global vectors for word representation (GloVe) [42] is designed to leverage both the local context window and the global statistical information of co-occurrence. The main idea of GloVe is to optimize the dot product between a center word and a neighboring word to be their co-occurrence probability in the corpus.

One can utilize the pretrained word embeddings in the music retrieval scenario as they form a semantically meaningful similarity space. It can alleviate the fixed vocabulary issue by retrieving the nearest music tag in the word embedding space. Also, the pretrained embeddings can be projected to form a multimodal embedding space with music audio [90, 91]. Details of the multimodal embedding space for tag-based music retrieval are described in Section 5.2.

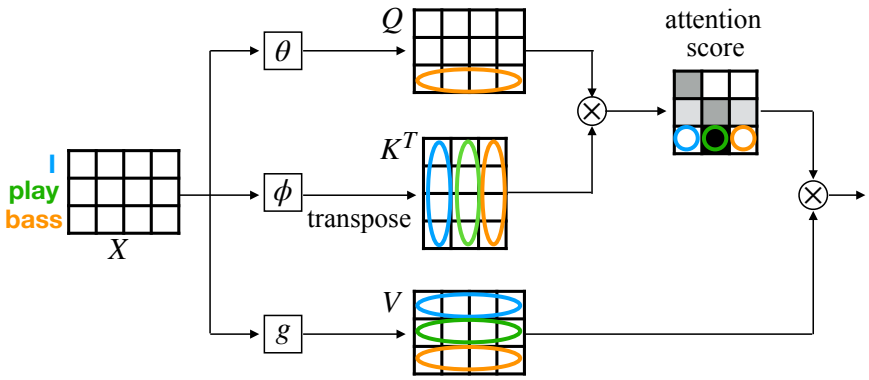


Figure 2.14: Self-attention mechanism.

2.4.3 Transformers

Self-attention

Sequence modeling is an important topic in natural language processing, speech recognition, and music representation learning, as their input formats are sequential. Recurrent neural network (RNN) [92] has been broadly used in speech recognition [93, 94], machine translation [95], and music tagging [96]. However, RNN has an inherent issue: a vanishing gradient problem. In RNN, the current hidden state results from the previous hidden states. As the sequence gets longer, it gets more difficult to learn the long-range dependency. Unlike RNN, a self-attention module [38] computes the response at a location in a sequence by attending to all locations within the same sequence. As a result, the self-attention mechanism has become a substitute for RNN to capture a long-range structure within sequential data.

There are three core concepts in self-attention mechanism: query, key, and value. From the given query (Q), the machine learns the relation between the query and keys (K) to compute the attention scores, and multiply the attention scores to the values (V). Finally, the sum of the attended values composes the semantics of the given query. For example, there is

a sentence "I play bass". With *bass* alone, we don't know if it is a fish or an instrument. We know it is an instrument based on the context because it has *play* in the sentence. When we want to know the semantic of *bass* (Q), we calculate the attention score by comparing the distance between *bass* and other words (K) in the sequence: *I*, *play*, and *bass*. This process is formalized as follow:

$$\begin{aligned} Q &= \theta(X) = XW_\theta \\ K &= \phi(X) = XW_\phi \\ V &= g(X) = XW_g \end{aligned} \tag{2.11}$$

where $\theta(\cdot)$, $\phi(\cdot)$, $g(\cdot)$ are learnable transformations, and *W* are their weights. As shown in Figure 2.14, word embeddings are transformed into different vectors. Then query vectors (Q) and key vectors (K) are multiplied using dot product to calculate the attention score. In the figure, dot product between *bass* (orange circle in Q) and all words (blue, green, and orange circles in K) are calculated to form attention scores. Blue circle in attention score indicates the attention score that *I* contributes to *bass*, green circle indicates *play* to *bass*, and orange circle indicates the attention score of *bass* to *bass*. In this context, for a given query *bass*, *play* will have higher attention score than *I* since *play* is a more important component to make *bass* as an instrument. Finally, we multiply the values (V) with the attention scores to calculate the output embeddings. In the output embedding vectors, each word embedding includes the context. The calculation of self-attention is formalized as follow:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.12}$$

where d_k is a dimension of keys and Q, K, V are matrices whose shapes are $Sequence \times Embedding$.

Transformer

Transformer[38] is a representation model relying entirely on an attention mechanism to learn global dependencies between input and output.

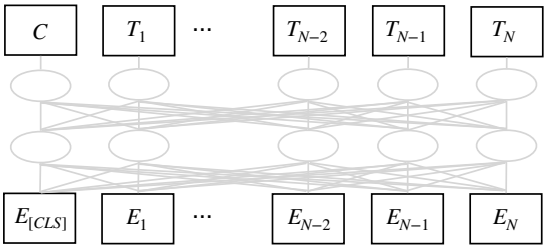


Figure 2.15: BERT architecture. Lower is input and upper is output.

By stacking multi-head self-attention layers, the transformer could dispense convolutional layers and recurrent layers, and established a new single-model state of the art in machine translation tasks. While the original transformer [38] has encoder and decoder parts to tackle the machine translation task, the bidirectional encoder representations from transformers (BERT) [30] only uses encoders of the transformer.

Training process of BERT is self-supervised learning. There are two different pretext tasks to solve. The first is masked language modeling (MLM). Some percentage of the input word tokens are randomly masked, and the model is optimized to predict the masked tokens. During the training, 15% of the token positions are randomly selected. 80% of them are masked with $[MASK]$ token, 10% are replaced with random tokens, and 10% are unchanged. Another task is next sentence prediction (NSP). This task is designed to understand the relationship between two sentences. The machine has to predict whether two sentences are actual continuous pair or not. For each sentence, the actual next sentence is chosen for 50% of the time, and a random sentence is selected for the rest 50%.

As shown in Figure 2.15, BERT is a stack of self-attention layers. In the first time step, there is a special $[CLS]$ token that summarizes the entire sequence to perform sequence classification. Also, it can perform token-level classification by returning predictions at each time step. Pretrained BERT is fine-tuned with labeled data. This versatile architecture obtained new state-of-the-art results on eleven NLP tasks [30].

2.5 Summary

Through this chapter, we reviewed relevant concepts and previous works that are crucial in this dissertation research. Section 2.2 discussed key concepts of music classification, including the task descriptions, input representations, evaluation metrics, and instance-level training. These will guide Chapter 3 to design and evaluate new music representation models properly. Section 2.3 reviewed various training schemes beyond supervised learning, such as transfer learning, semi-supervised learning, and self-supervised learning. They enable the inclusion of large-scale external or unlabeled data during the training process. These training schemes are introduced for music representation learning in Chapter 4 to enhance models' generalizability. Finally, recent updates of natural language processing are reviewed in Section 2.4. Pretrained word embeddings can enable flexible vocabulary for music retrieval in Section 5.2. And sentence-/paragraph-level text representations can be bridged together with music audio representations to form multimodal music representation spaces in Section 5.3.

Chapter 3

MUSIC REPRESENTATION LEARNING

3.1 Introduction

Representation learning [28] is a set of machine learning techniques that automatically learns representations of the data. The learned representations are further utilized for classification or prediction. In traditional machine learning approaches, this step has been done by humans. Researchers manually designed ingenious features based on their domain knowledge. On the other hand, modern deep learning approaches alternate the process by letting the machine automatically learn relevant features from the data. For example, we want to build a classifier that determines whether a song delivers *happy* mood or not. In traditional approaches, we need to design features that are relevant to keys and tempo since we think those features are important for predicting the mood *happy* (e.g., a major key with a fast tempo). However, in representation learning, we don't need the manual feature design process. From the data labeled with *happy* and *not happy*, the machine will automatically learn relevant features for the classification in an end-to-end way. Furthermore, with self-supervised representation learning approaches [78, 97], we do not need labels but only data.

By leveraging large-scale data, deep representation learning has firmly established the state-of-the-art in many domains including computer vision (CV) [79, 98], natural language processing (NLP) [30, 99], and music information retrieval (MIR). MIR researchers have adopted the successful deep representation models to solve a variety of problems including beat tracking [100], pitch estimation [101], instrument identification [102], mood classification [103], and automatic music tagging [104]. Especially, automatic music tagging is one of the most actively explored areas in MIR using deep representation learning due to the availability of scalable datasets [54, 43, 40] which are rare in MIR research. Since music tags cover multiple facets of music characteristics and their scalable datasets suit data-driven research, in this chapter, we explore automatic music tagging as a proxy of music representation learning.

This chapter is organized as follows. Section 3.2 introduces datasets and evaluation of the automatic music tagging task. Section 3.3 tackles the problem of heterogeneous experimental setups and revisit existing music tagging models under a homogeneous evaluation pipeline. Section 3.4 presents data-driven front-end filters for music representation learning. Section 3.5 introduces the transformer [38, 30] to music tagging and shows its versatility and interpretability. Finally, section 3.6 summarizes the work and provides some links with the next chapter.

This chapter includes the following works:

- Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra, Evaluation of CNN-based Automatic Music Tagging Models, Sound and Music Computing (SMC) 2020.
- Minz Won, Sanghyuk Chun, Oriol Nieto, and Xavier Serra, Data-driven Harmonic Filters for Audio Representation Learning, The International Conference on Acoustics, Speech, Signal Processing (ICASSP) 2020.
- Minz Won, Sanghyuk Chun, and Xavier Serra, Toward Interpretable Music Tagging with Self-attention, ArXiv 2019.
- Minz Won, Keunwoo Choi, and Xavier Serra, Semi-supervised Music Tagging Transformer, The International Society for Music Information Retrieval (ISMIR) 2021.

3.2 Automatic music tagging

3.2.1 Datasets

MagnaTagATune (MTAT) [54] is one of the most commonly used datasets for benchmarking automatic music tagging systems. It contains multi-label annotations by genre, mood, and instrumentation for 25,877 audio segments, each 30s long. The data was collected using the TagATune game, where participants have to answer whether each tag is relevant to each audio segment. Note that some segments are from the same song, which means the number of unique songs is smaller than the number of segments. The audio is in the MP3 format (32 Kbps bitrate and 16 kHz sample rate). Originally the dataset is split into 16 folders, and commonly the first 12 folders are used for training, the 13th for validation, and the last three are used for testing. Only 50 most frequent tags are typically used for the task.

Million Song Dataset (MSD) [43] is a dataset of audio features for one million songs, partially expanded by the MIR community with crowd-sourced tags from *Last.fm* as well as a mapping to 30s audio preview segments originally obtained from *7digital*.[1] In total, this subset of the dataset contains 241,904 annotated song segments and it is commonly used for benchmarking music tagging models on a larger scale. The tags cover genre, instrumentation, moods and decades. The audio segments vary in quality, being encoded as MP3s with a bitrate from 64 to 128 Kbps and sample rate of 22 kHz or 44 kHz. Similar to MTAT dataset, most previous works use only 50 most frequent tags. Note that the tag annotations available for this dataset are inherently noisy as they come from a free-form social tagging application for music enthusiasts and are used without any preprocessing intended to improve the quality of tags [105].

MTG-Jamendo Dataset [40] contains audio for 55,701 full songs and is built using music publicly available on the *Jamendo*[2] music platform under Creative Commons licenses. The minimum duration of each song

[1] https://www.7digital.com

[2] https://jamendo.com

is 30s, and they are provided in the MP3 (320 Kbps bitrate). Thus, this dataset contains significantly larger audio segments with higher encoding quality than MTAT and MSD. The tracks in the dataset are annotated by 692 different tags covering genres, instrumentation, moods and themes. All tags were originally provided by the artists submitting music to Jamendo, but they were preprocessed with the goal of tag cleaning by the creators of the dataset. Different from MTAT and MSD, official splits of the data are provided for training, validation and test. There are splits for top50-tagging, genre classification, mood / theme classification, and instrument classification.

3.2.2 Evaluation

Most automatic music tagging research uses the area under receiver operating characteristic curve (ROC-AUC) as their main evaluation metric. However, the ROC-AUC score can be overly optimistic when it is applied to imbalanced (highly skewed) data [59]. Music tagging researchers report the area under precision-recall curve (PR-AUC) together with ROC-AUC scores on this account. Details of ROC-AUC and PR-AUC are described in Section 2.2.4. The ground truth tag labels exist in a song-level but some music tagging models make instance-level predictions due to the limited size of receptive fields (Section 2.2.5). We need to aggregate the instance-level predictions to generate song-level predictions so that we can assess the model's performance (measure ROC-AUC and PR-AUC). Following most previous works, in this dissertation, we average the instance-level predictions to generate song-level predictions when the model is trained with instance-level training.

Music tagging is a multi-label binary classification task. If we tackle the task with 50 tags, there are 50 AUC scores. But it is preferable to have a single representative value when we compare multiple models. The option "micro" calculates the metrics globally, "macro" averages the tag-level metrics, and also it is possible to give different weights in average process to take data imbalance into account. In this thesis, all evaluation metrics are averaged using "macro" following most previous works.

3.3 Comparison of CNN-based models

3.3.1 Motivation

To tackle the problem of music tagging, recent studies in MIR adopted deep neural networks, mostly based on convolutional neural networks (CNNs) [29]. The introduction of CNN helped to break the previous glass ceiling in the performance of music tagging systems and researchers started actively proposing their own architecture design. As a result, the hand-crafted feature-based approaches were replaced by data-driven feature learning approaches in most recent automatic music tagging research. However, unfortunately, it is difficult to compare the proposed architectures directly with each other due to their different experimental setups when reporting results (e.g., dataset splits, library versions, computing environments, and optimization methods). Furthermore, the related information is sometimes unclear on the paper that results in unintentional re-using and comparison of previous reports which are incompatible performance values. In this section, we address this issue and report experimental results for various state-of-the-art music tagging models using three different datasets (MagnaTagATune, Million Song Dataset, and MTG-Jamendo dataset) with a consistent experimental setup. In addition, we conduct experiments to assess the robustness of these architectures against four different types of deformations [106] and determine their generalization abilities.

3.3.2 Inconsistent experimental setups

Data split

Different library versions, computing environments, and optimization methods may lead to inconsistent experimental results. One of the most critical issues of current music tagging research is inconsistent data splits. This information is not clearly explained in each paper which results in unintended wrong baselines. We could reproduce previous works after collecting the details by contacting each author.

In MTAT dataset, most previous works use top 50 tags as described in previous section. That means some audio clips may not have any of those 50 tags and this can affect training and evaluation of music tagging. A group of researchers [33, 107] used the dataset as it is (≈26k clips). Another group of researchers [57, 58, 108] discarded the clips without any tags (results in ≈21k clips). This results in totally different scores even if they use the same data split.

Inconsistent split also exists in the MSD tagging. Originally, the MSD does not include audio. However, audio preview segments can be collected from the web by mapping their metadata. Since the audio segments are collected, not designed to be a dataset, they have different lengths. Different from most tagging models that utilize instance-level training [107, 57], FCN [33] performs song-level training, hence all audio inputs need to be the same length. To this end, the authors of FCN discarded audio segments shorter than 29.1s.

Optimization techniques

Optimization is another critical part of deep learning experiments. Sometimes it is more critical than the model architecture. Various optimization techniques have been introduced in machine learning research, such as stochastic gradient descent (SGD), Adam [109], AdamP [110], and AdamW [111]. Since previous music tagging works use different optimization techniques, it is impossible to compare those representation models with the reported metrics. There is a possibility that the optimization trick is more critical than the model architecture.

Preprocessing

Different preprocessing may affect the performance as well. Each previous work used different sampling rate (12kHz [33, 96], 16kHz [112, 57, 108], 22kHz [113, 80]), different short-time Fourier transform (STFT) parameters, and different numbers of mel filterbanks (96 or 128).

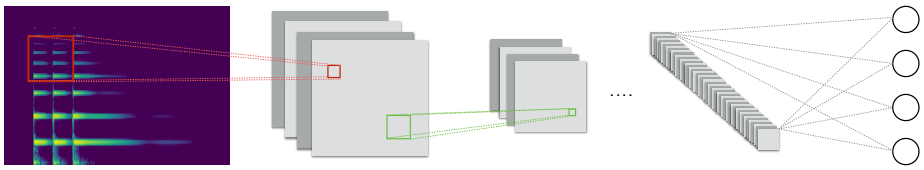


Figure 3.1: Fully convolutional network

Our solution

In this work, we reproduce previous music tagging models under the same experimental setup. This holistic evaluation of music tagging models will enable the fair comparison of music representation models so that researchers can choose the appropriate models for their purposes. We use three different datasets: MTAT [54], MSD [43], and MTG-Jamendo dataset [40]. For MTAT tagging, we use the tracks with at least one top-50 tag. This is the same setting from [57, 58, 108] which includes ≈21k clips.[3] For MSD tagging, we follow the dataset split commonly used by music tagging researchers.[4] This split includes 201,680 songs for training, 11,774 for validation and 28,435 for testing. We used a mixture of SGD and ADAM for optimization which was introduced in [114]. For audio preprocessing, 16kHz sampling rate, 512-point FFT with 50% overlap, and 128 mel bands are used.

3.3.3 Music representation models

This subsection introduces various music representation models that we used for the holistic evaluation.

Fully convolutional network (FCN)

A fully convolutional network (FCN) [115] is a variant of CNN that consists of only convolutional layers without any fully-connected layers. A

[3] https://github.com/jongpillee/music_dataset_split/tree/master/MTAT_split
[4] https://github.com/jongpillee/music_dataset_split/tree/master/MSD_split

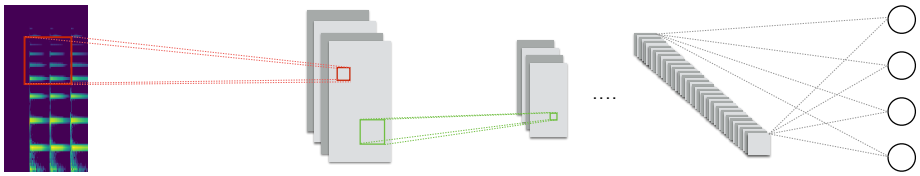


Figure 3.2: VGG-ish or short-chunk CNN with instance-level training.

FCN for music tagging uses mel spectrogram inputs. In the preprocessing step, a 29.1s audio segment is converted to a 96×1366 mel spectrogram. It is then used as an input and is passed through 4 convolutional layers. Each convolutional layer uses homogeneous 3×3 2D filters followed by a max-pooling layer (Figure 3.1). Different sizes of strides are used for max-pooling layers ((2, 4), (4, 5), (3, 8), (4, 8)) to increase the size of receptive fields to cover the entire input mel spectrogram (96×1366). In the original paper, FCN was trained with a song-level training method since the track durations in MTAT dataset correspond to the size of the receptive field. However, this is not the case for MTG-Jamendo dataset containing longer tracks, where instance-level (29.1s) training is applied.

Short-chunk CNN

According to the previous work [108], a simple 2D CNN with 3×3 filters can already claim exceptional results when it is trained with short chunks of audio, i.e., instance-level training. It is a very prevalent type of CNN (sometimes referred to as *vgg*-like, see Figure 3.2) but, to the best of our knowledge, there are no references for this architecture design in music tagging research. Hence, we implemented a 7-layer CNN with a fully-connected layer, and its extension with residual connections [79]. Different from FCN, it uses a smaller size of max-pooling (2×2) because the input segment is way shorter than the song-level inputs (29.1s). We used 128 mel bins so that 7 max-pooling layers can summarize them into a single dimension ($2^7 = 128$). It uses 3.69s audio excerpts, hence we call this model "short-chunk CNN" in this work to differentiate it from FCN.

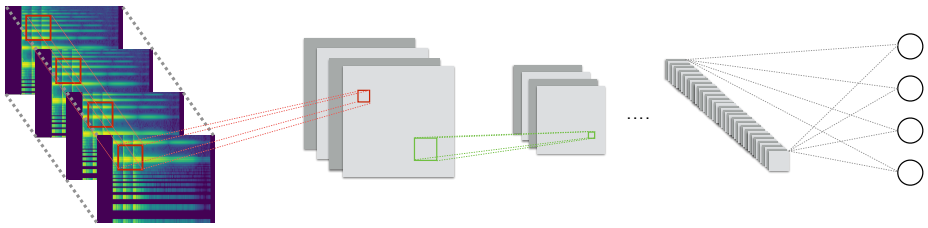


Figure 3.3: Harmonic CNN

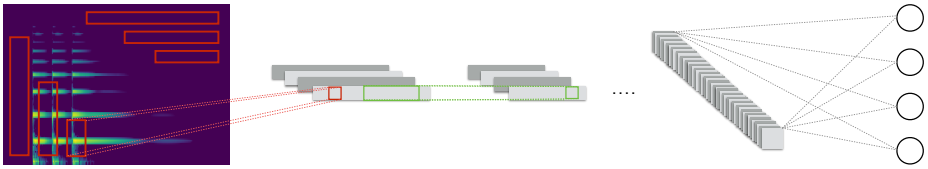


Figure 3.4: MusiCNN

Harmonic CNN

Harmonic CNN [108] takes advantage of trainable band-pass filters and harmonically stacked time-frequency representation inputs. Trainable filters (mainly trainable bandwidths) bring more flexibility to the model. And harmonically stacked representation preserves spectro-temporal locality while keeping the harmonic structures through the channel of the input tensor in the first convolution layer (Figure 3.3) as introduced in [116]. The number of trainable frequency bands is set to 128 and the number of harmonics considered for stacking is 6. Instance-level training with 5s audio segments is performed. More details are described in the next section (Section 3.4).

MusiCNN

The MusiCNN [112] model also uses mel spectrograms as its inputs. The architecture design choices in MusiCNN rely on some intuition from the music domain knowledge. The first convolutional layer of MusiCNN

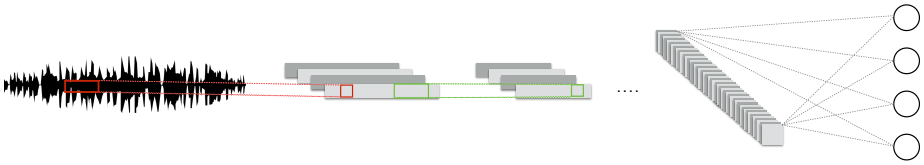


Figure 3.5: Sample-level CNN

consists of vertical and horizontal filters. Vertical filters are designed to capture pitch-invariant timbral features (bottom-left of Figure 3.4): e.g., 38×7 filter captures sub-band information of short period of time. To enforce the pitch-invariancy, the following max-pooling layer pools the maximum value across the frequency axis. Horizontal filters, on the other hand, capture temporal energy envelope of the audio. After the mean-pooling across the frequency axis of input mel spectrograms, horizontally long filters (e.g., 1×165) capture the temporal energy patterns (top-right of Figure 3.4). The extracted timbral and temporal features are concatenated through the channel, then the following 1D convolutional layers summarize them to predict relevant tags. Different from FCN, the MusiCNN only uses short audio excerpts (3s) as its inputs during training, i.e., instance-level training.

Sample-level CNN

Sample-level CNN [57] tackles the automatic music tagging problem in an end-to-end fashion (Figure 3.5). It takes raw audio waveforms as its inputs. Sample-level CNN is simpler and deeper than mel-spectrogram-based approaches. It consists of ten 1D convolutional layers with 1×3 filters and 1×3 max-poolings. Trained front-end filters perform similar to the process of deriving mel spectrograms and the back-end convolution layers summarize the sequence of the extracted features. We also considered a variation of sample-level CNN [58] with squeeze-and-excitation (SE) [117] blocks. Sample-level CNN and its variant with SE blocks also use short audio excerpts (3.69s) for the instance-level training.

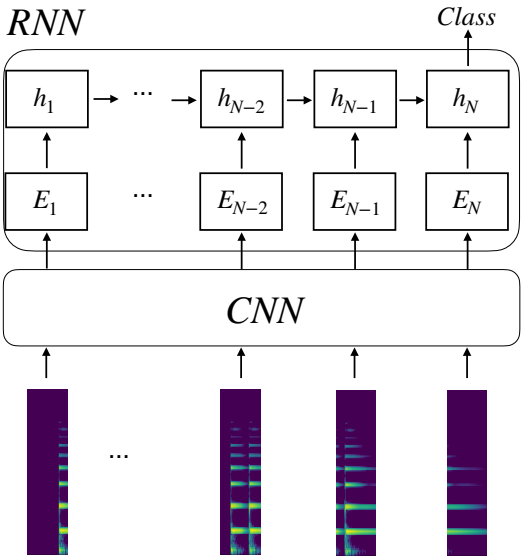


Figure 3.6: Convolutional recurrent neural network (CRNN)

Convolutional recurrent neural network (CRNN)

Convolutional recurrent neural network (CRNN) [96] uses mel spectrogram inputs. A CRNN can be described as a combination of CNNs and RNNs. The CNN front end extracts local features and the RNN back end summarizes them temporally (Figure 3.6). Since RNNs are more flexible than CNNs for summarizing sequential information, it can be beneficial to use RNNs for predicting tags that may be affected by global structures (e.g., moods/themes). Four convolutional layers with 3×3 2D filters are used in the front end and two-layer RNNs with gated recurrent units (GRU) are used in the back end. Long music excerpts (29.1s) are used as inputs of CRNN. In other words, it performs song-level training for MTAT and MSD, and instance-level training for MTG-Jamendo dataset.

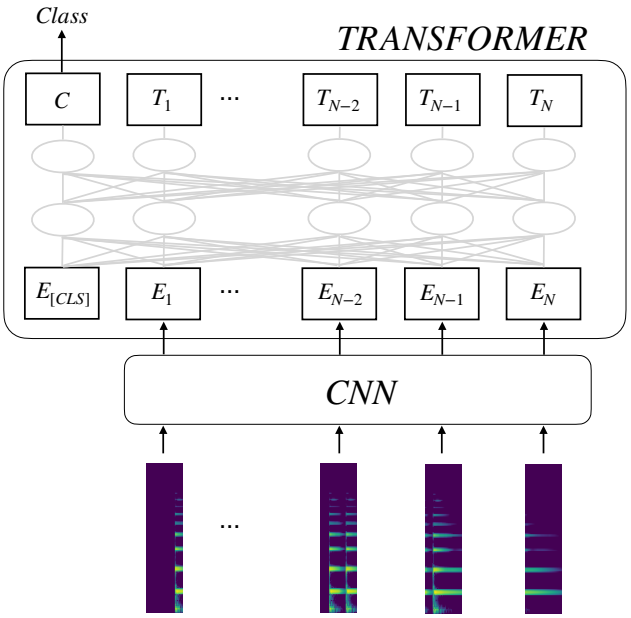


Figure 3.7: Convolutional neural network with self-attention (CNNSA) or music tagging transformer.

Convolutional neural network with self-attention (CNNSA)

The self-attention-based music tagging model shares the same intuition as CRNN to extract local features with CNNs and summarize them with sequence models. The only difference is that the self-attention mechanism is used instead of the RNNs for the temporal summarization back end (Figure 3.7). Motivated by its huge success in natural language processing [30], we adapted the Transformer encoder, which is a deep stack of self-attention layers, for automatic music tagging. 15s-long audio excerpts are used for training CNNSA. More details of CNNSA and its advanced version are described in Section 3.5.

Methods	MTAT		MSD		MTG-Jamendo	
	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
FCN [33]	0.9005	0.4295	0.8744	0.2970	0.8255	0.2801
FCN (with 128 Mel bins)	0.8994	0.4236	0.8742	0.2963	0.8245	0.2792
MusiCNN [107]	0.9106	0.4493	0.8803	0.2983	0.8226	0.2713
MusiCNN (with 128 Mel bins)	0.9092	0.4546	0.8788	3036	0.8275	0.2810
Sample-level [57]	0.9058	0.4422	0.8789	0.2959	0.8208	0.2742
Sample-level + SE [58]	0.9103	0.4520	0.8838	0.3109	0.8233	0.2784
CRNN [96]	0.8722	0.3625	0.8499	0.2469	0.7978	0.2358
CRNN (with 128 Mel bins)	0.8703	0.3601	0.8460	0.2330	0.7984	0.2378
CNNSA [114]	0.9077	0.4445	0.8810	0.3103	0.8261	0.2883
Harmonic CNN [108]	0.9127	0.4611	**0.8898**	**0.3298**	0.8322	0.2956
Short-chunk CNN	0.9126	0.4590	0.8883	0.3251	**0.8324**	**0.2976**
Short-chunk CNN + Res	**0.9129**	**0.4614**	**0.8898**	0.3280	0.8316	0.2951

Table 3.1: Performances of CNN-based music tagging models.

3.3.4 Performance comparison

We report ROC-AUC and PR-AUC of all implemented models using three datasets in Table 3.1. In general, models trained with short audio excerpts (MusiCNN, variants of sample-level CNN, CNNSA, Harmonic CNN, variants of short-chunk CNN) outperform other models trained with relatively longer audio segments (FCN, CRNN). Training with short chunks (instances) is noisier: e.g., an audio excerpt can have a tag *guitar* if a guitar appears in the song even though the selected excerpt doesn't include guitar sound in it. However, one can expect a much larger number of examples during the training (e.g., 25,877 tracks $\times$ 16 chunks = 414,032 examples). We suspect this brings the performance gain when the model is trained with short instance-level examples. Furthermore, most of the top 50 tags in the three datasets can be identified only with a short audio excerpt (e.g., instruments, genres). Thus, the model does not need a long sequence of audio to perform its binary classification task. For the top 50 tags in each dataset we experimented with, it is more beneficial to use instance-level training with short audio excerpts than the song-level training.

Short-chunk CNN, short-chunk CNN with residual connections, and Harmonic CNN showed the best results for every dataset. These three models are trained on short audio excerpts (3.69s or 5s) and they use 3×3 convolutional filters followed by 2×2 max-poolings. FCN uses similar filters, but with larger max-poolings which increase its size of the receptive field to fit long audio segments (29.1s). We conclude that smaller max-poolings with shorter audio excerpts work better for CNNs with 3×3 filters.

MusiCNN shows competitive results in MTAT. However, other models (sample-level + SE, CNNSA) outperform MusiCNN on larger datasets (MSD and MTG-Jamendo). This confirms an intuition that domain knowledge can be beneficial for relatively small datasets, reported in [112]. However, the design choices of MusiCNN restricts the power of the model when it is trained with larger datasets.

For the sequential models, CNNSA outperforms the CRNN. Different

from self-attention mechanisms, RNNs with long sequence inputs suffer from vanishing gradient problems. Self-attention mechanism alleviates the problems by providing direct paths between all time steps. According to the reported visualizations [114], self-attention performs well for pinpointing relevant short-time acoustic features in the audio sequence, but it was difficult to determine if the model learned long-time characteristics properly. To determine such abilities, some tags related to a global structure have to be cherry-picked and evaluated.

Since FCN, MusiCNN, and CRNN use mel spectrogram inputs with 96 mel bands, there can be relative disadvantages when they are compared with other models using 128 mel bands. For the fair comparison, we experimented with FCN, MusiCNN, and CRNN using 128 mel bands. A larger number of mel bands did not show any significant impacts on the performances. Since each architecture design was optimized for a smaller number of mel bands, simply increasing the size of input mel bands cannot guarantee the optimized performance of the models.

3.3.5 Robustness studies

Input deformations

To further investigate the performance of different state-of-the-art models, we conducted robustness studies. If a pretrained model has good generalization abilities, the prediction of the model should not be sensitive against small perturbations in the input audio. By applying four different audio deformations to the test set (pitch shift, time stretch, dynamic range compression, and addition of white noise), we intended to determine the generalization abilities of the models. Note that we applied these four deformations only to the test set, which means that the models have never been exposed to the same deformations during training. All employed deformations are based on an existing music data augmentation framework (MUDA)[5] [106]:

- **Pitch shift** by $n \in \{-1, 1\}$ semitones.

[5] https://github.com/bmcfee/muda

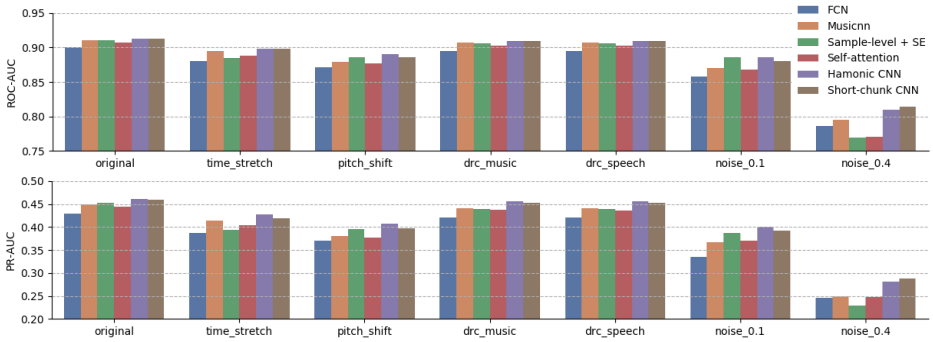


Figure 3.8: Evaluations metrics with perturbed audio inputs. Dynamic range compression is shortened as "drc" in the plot.

- **Time stretch** by $\gamma \in \{2^{-1/2}, 2^{1/2}\}$.
- **Dynamic range compression** following *speech* and *music (standard)* settings of Dolby E standards [118].
- **White noise addition** $x_{mixed} = (1 - \alpha) \cdot x + \alpha \cdot x_{noise}$ where $\alpha \in \{0.1, 0.4\}$.

Robustness results

Figure 3.8 shows performances of each model under various input deformations. Here we tested FCN, MusiCNN, sample-level + SE, self-attention, Harmonic CNN, and short-chunk CNN. We followed the original input preprocessing of each model because a larger of mel bands did not show significant effects in our experiment. CRNN is not included due to its relatively low performance.

Dynamic range compression was the least influential and the white noise addition (0.4) was the most critical among the four different perturbations considered. MusiCNN is robust against time stretching but it is relatively vulnerable against pitch shift. We suspect the max pooling layer over frequency axis hinders the MusiCNN from learning generalized representations. Harmonic CNN and short-chunk CNN were the two

best models with original data. However, Harmonic CNN showed better generalization abilities against input deformations except for the white noise addition (0.4). Sample-level CNN with SE blocks showed good performance with a small amount of noise (0.1), but it could not generalize when this amount was increased (0.4).

3.3.6 Conclusion

In this section, we revisit state-of-the-art automatic music tagging models and report their performances with a consistent experimental setup. In general, short-chunk-based approaches (instance-level training) showed better results than models trained with larger input segments (FCN, CRNN). The design choices followed by MusiCNN could show good performance on a small dataset, but it restricted the model from learning more information on larger datasets. Sequential models (CRNN, CNNSA) showed competitive results but could not outperform other models since most of tags in the datasets do not require long sequences for their identification. Interestingly, the best performing model is a simple CNN with 3×3 filters trained on short audio excerpts (short-chunk CNN). Although the original design choice of the CNN is from computer vision, it outperformed other methods except for Harmonic CNN.

We further assessed generalization abilities of models by testing perturbed inputs. We could observe a different ranking of the models in terms of their performance on each deformation. In our experiment, Harmonic CNN and short-chunk CNN consistently report better scores than other models. Specifically, Harmonic CNN showed the best generalization abilities against every deformation types except for a heavy white noise addition. Since the models cannot generalize to unseen type of deformations, the efficacy of data augmentation in music tagging has to be further investigated.

3.4 Data-driven harmonic filters

3.4.1 Introduction

With the emergence of deep learning, end-to-end data-driven approaches have become prevalent in audio representation learning [119]. Domain knowledge is often de-emphasized in modern deep architectures and is minimally used in preprocessing steps (e.g., mel spectrograms). Recent works, with no domain knowledge in their architecture design and preprocessing, reported remarkable results in automatic music tagging [57], voice search [120], and environmental sound detection [121], by using raw audio waveforms directly as their inputs.

Nevertheless, we believe that domain knowledge may facilitate more efficient representation learning, especially when the amount of data is limited [112]. Given that harmonic structure plays a key role in human auditory perception [122], we present a model with a front end module that can learn compelling representations in a data-driven fashion while forcing the network to employ such harmonic structures. This front-end module, which we call Harmonic filters, is a trainable filter bank [123, 124, 125, 126, 127] that preserves spectro-temporal locality with harmonic structures [116]. Thus, these Harmonic filters aim to bridge the modern assumption-free approaches with the traditional hand-crafted techniques, with the goal to reach a "best of both worlds" scenario.

Contribution. Our contribution is three-fold: (*i*) we propose a versatile front-end module for audio representation learning with a set of data-driven harmonic filters, (*ii*) we show that the proposed method achieves state-of-the-art performance in three different audio tasks, and (*iii*) we present analyses on the parameters of our model that depict the importance of harmonics in audio representation learning.

Organization. This section is organized as follows: We introduce the Harmonic filters and their architecture design in Section 3.4.2. Section 3.4.3 describes the tasks and datasets used to assess the Harmonic filters. Section 3.4.4 reports experimental results and analyses. Finally, we draw conclusions and discuss future work in Section3.4.5.

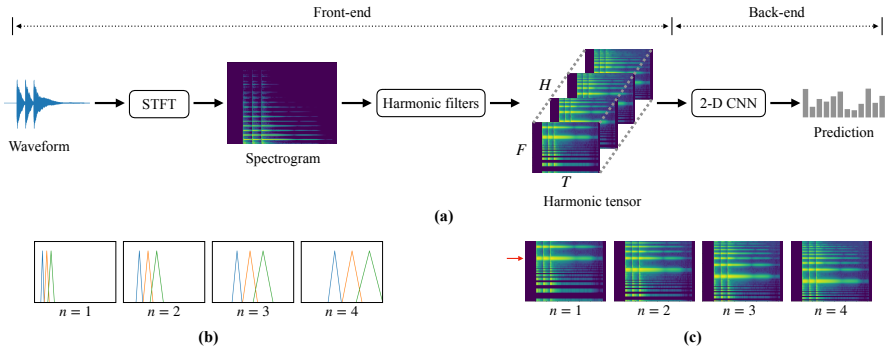


Figure 3.9: (a) The proposed architecture using Harmonic filters. The proposed front-end outputs the Harmonic tensor and the back-end processes it depending on the task. The Harmonic filters and the 2-D CNN are data-driven modules that learn parameters during training. (b) Harmonic filters at each harmonic. (c) An unfolded Harmonic tensor. The red arrow indicates the fundamental frequency.

3.4.2 Architecture

Previous Harmonic Representations

The harmonic constant-Q transform (HCQT) [116] is a 3-dimensional representation whose dimensions are *harmonic* (H), *frequency* (F), and *time* (T). By stacking standard constant-Q transform (CQT) representations, one harmonic at a time, the output representation (i.e., HCQT) can preserve the harmonic structure while having spectro-temporal locality. A fully convolutional neural network (CNN) with HCQT inputs could achieve state-of-the-art performance in multi-f0 and melody extraction tasks using several datasets [116].

In our previous work [128], we used two learnable sinc functions (i.e., $\sin(x)/x$) to form each band-pass filter of the first convolutional layer [127], such that the set of harmonics can be learned. By aligning the convolution band-pass filters in each *harmonic*, the first layer outputs an $H \times F \times T$ tensor. When the first harmonic center frequencies are initialized with a MIDI scale, this can be interpreted as an extended, more

flexible version of HCQT.

However, the convolution band-pass filter approach to get harmonic spectro-temporal representations requires many convolutions ($H \times F$), including redundant ones (e.g., a 440Hz filter is equivalent to the second harmonic filter of 220Hz). To overcome these efficiency limitations, in this work we replace the convolution band-pass filters of our previous work with an STFT module followed by learnable triangular filters, the so-called Harmonic filters.

Harmonic Filters

First, we formulate a triangular band-pass filter Λ as a function of a center frequency f_c and a bandwidth BW as follows:

$$\Lambda(f; f_c, BW) = \left[1 - \frac{2|f - fc|}{BW}\right]_+, \tag{3.1}$$

where $[\cdot]_+$ is a rectified linear function, and f is the frequency bin. Note that when there are multiple triangular band-pass filters with mel scaled center frequencies, the filter bank performs similarly to the mel filter bank.[6]

Empirically, the bandwidth BW can be approximated as an affine transform of f_c: $BW \simeq 0.1079 f_c + 24.7$ (equivalent rectangular bandwidth (ERB) [129]). For flexibility's sake, we let the data decide the affine transform with parameters α, β, and Q: $BW = (\alpha f_c + \beta)/Q$.

Now, we define a Harmonic filter Λ_n as follows:

$$\Lambda_n(f; f_c, \alpha, \beta, Q) = \left[1 - \frac{2|f - n \cdot f_c|}{(n \cdot \alpha f_c + \beta)/Q}\right]_+. \tag{3.2}$$

The Harmonic filter Λ_n is a triangular band-pass filter of the n-th harmonic of center frequency f_c. Then, our proposed filter bank is defined as a set of Harmonic filters as follows:

$$\{\Lambda_n(f; f_c) \mid n = 1, \ldots, H, f_c \in \{f_c^{(1)}, \ldots, f_c^{(F)}\}\}, \tag{3.3}$$

[6] It is not equivalent because mel filters have asymmetrical triangle shapes.

where $f_c^{(i)}$ denotes the i-th center frequency in the first harmonic. Figure 3.9-(b) shows Harmonic filters with $H = 4$ and $F = 3$. Bandwidths go wider as center frequencies go higher.

Note that, for a given input spectrogram, when $H = 1$ and f_c are initialized with a mel scale, the filter bank will return an output analogous to the mel spectrogram. When $H > 1$, we stack the outputs aligned with *harmonic* so that we can have a tensor of dimensionality $H \times F \times T$ as shown in Figure 3.9-(a). We call this 3-dimensional tensor a Harmonic tensor. Exploiting locality in *time*, *frequency*, and *harmonic* by using this type of representation is advantageous, as discussed in [116]. Furthermore, this Harmonic tensor is flexible since the center frequencies f_c and the bandwidth parameters α, β, Q are all learnable in a data-driven fashion.

Back-end

Deep networks for audio representation learning can be divided into front end and back end: a feature extractor and a classifier, respectively [112]. Figure 3.9-(a) shows the overview of the proposed architecture. We use an STFT module followed by Harmonic filters as our front end. For the back end, a simple conventional 2-D CNN is used since our main goal is to emphasize the advantages of using learnable Harmonic tensors. Harmonics are treated as channels to be fed into the 2-D CNN, thus capturing the harmonic structure through each of its channels. This design choice enforces the convolutional filters to embed harmonic information with locality in time and frequency.

Figure 3.9-(c) shows an unfolded Harmonic tensor of a 440Hz piano sound. We indicate the fundamental frequency with a red arrow. From left to right, we can see the intensity of the first, second, third, and fourth harmonics at once.

Implementation details

First, harmonic center frequencies f_c of the Harmonic tensor are initialized to have a quarter tone interval: $f_c(k) = f_{min} \cdot 2^{k/24}$, where k is the fil-

ter index and $f_{min} = 32.7$Hz (C1) is the lowest frequency. The maximum frequency of the first harmonic f_{max} is defined as: $f_{max} = f_s/2H$, where f_s is the sampling rate. After the parameter study, we set the number of harmonics H to 6 for inputs with a 16kHz sampling rate. This results in 128 frequency bins ($F = 128$), with a total of 768 Harmonic filters.

The back end CNN consists of seven convolutional layers and one fully connected layer to predict the outputs. Each layer includes batch normalization [130] and ReLU nonlinearity. The final activation function is a sigmoid or a softmax, depending on the task. Models are trained for 200 epochs and we choose the best model based on the evaluation metric in the validation set. Scheduled Adam [109] and stochastic gradient descent (SGD) were used for stable convergence as proposed in [114].

3.4.3 Tasks and Datasets

To show the versatility and effectiveness of the Harmonic filters, we experiment with three different tasks: automatic music tagging, keyword spotting, and sound event tagging.

Automatic music tagging. We used the previously introduced MagnaTagATune (MTAT) dataset, and report ROC-AUC and PR-AUC — see Section 3.2. Many music tags such as genre, instrumentation, and moods are highly related to the timbre of audio, and harmonic characteristics are crucial for the timbre perception. Hence, one can expect improvements in music tagging by adopting the Harmonic filters in the front end.

Keyword spotting. MFCC have long been used as input to many speech recognition models because harmonic structure is known to be important for the speech recognition. We believe the Harmonic filters will bring faster convergence and performance improvement than conventional 2-dimensional representations (e.g., CQT, mel spectrogram). The Speech Commands dataset [133] consists of ≈106k audio samples with 35 command classes (e.g., "yes," "no," "left," "right") for limited-vocabulary speech recognition. Trained models are trivially evaluated with the classification accuracy of choosing one of the 35 classes.

Sound event tagging. The DCASE 2017 challenge [134] used a subset of

Methods	Music Tagging		Keyword Spotting	Sound Event Tagging	
	MTAT		Speech Commands	DCASE 2017	
	ROC-AUC	PR-AUC	Accuracy	F1 (0.1)	F1 (opt)
Musicnn [112]	0.9089*	0.4503*	-	-	-
Attention RNN [131]	-	-	0.9390	-	-
Surrey-cvssp [132]	-	-	-	-	0.5560
Sample-level [57]	0.9054	0.4422	0.9253	0.4213	-
+ SE [58]	0.9083	0.4500	0.9395	0.4582	-
+ Res +SE [58]	0.9075	0.4473	0.9482	0.4607	-
Proposed	**0.9141**	**0.4646**	**0.9639**	**0.5468**	**0.5824**

Table 3.2: Performance comparison with state-of-the-art. The numbers are averaged across 3 runs. '*' denotes reproduced result with our data split. F1 (0.1) and F1 (opt) denote F1-score measured by threshold value of 0.1 and optimized one, respectively.

the AudioSet [135] for the task 4: "large-scale weakly supervised sound event detection for smart cars." It consists of ≈53k audio excerpts with 17 sound event classes, e.g., train horn, car alarm, and ambulance siren. Acoustic events are non-music and non-verbal audio signals, which are expected to have more "inharmonic" characteristics. We are particularly interested in exploring the performance of the proposed model on such audio signals, and thus this task is an ideal candidate for our research. This is also a multi-label classification task and we evaluate it using the average of instance-level F1-scores.

3.4.4 Experimental results

Performance comparison

We compare the Harmonic tensor based 2D CNN with the state-of-the-art models of each task. All the experimental results are averaged after three runs. As shown in Table 3.2, our model outperforms previous results in every task.

In music tagging, we reproduced Musicnn [112] with the same data cleaning and split strategy from others [57, 58] for a fair comparison. As a result, the mel spectrogram based approach [112] and the raw audio based approach [58] yield comparable results on the MTAT dataset. Our proposed model shows improvements from previous approaches in terms of ROC-AUC and PR-AUC.

As we expected, the keyword spotting accuracy of the proposed model is superior to previous works. Moreover, this showed remarkably fast convergence: the best model according to the validation loss was around 10 epochs while other tasks needed over 100 epochs.

The Harmonic filters were also effective when operating on relatively inharmonic audio signals. We report two different metrics for the DCASE 2017 dataset. F1 (0.1) indicates the F1-score when the threshold of prediction is 0.1, and F1 (opt) is the post threshold optimization score. Note that our model is superior to the state-of-the-art without data balancing or ensembles.

H	1	2	3	4	5	6	7*
ROC-AUC	0.9132	0.9115	0.9118	0.9118	0.9129	0.9141	**0.9146**
PR-AUC	0.4599	0.4541	0.4550	0.4555	0.4562	**0.4646**	0.4617

Table 3.3: The effect of number of Harmonics (H) on MTAT. '*' has a different size of max pooling due to the smaller F.

Options	512 FFT	256 FFT	Quarter tone	Semi tone
Q(MTAT)	2.1386	1.9537	2.1386	1.8447
Q(Speech Commands)	1.9032	1.9983	1.9032	1.8451
Q(DCASE 2017)	1.9040	1.8762	1.9040	1.8460

Table 3.4: Trained bandwidth parameter Q in different settings.

Parameter study

Here, we provide further understanding of the Harmonic filters by a parameter study and a qualitative analysis on the trained models.

We conduct the parameter study using the MTAT dataset to investigate how the number of harmonics H impacts performance. Table 3.3 summarizes the results. When $H = 1$, the Harmonic tensor is a 2-dimensional representation like a mel spectrogram or a CQT, but with frequency bins and bandwidth parameters that are automatically learned and initialized as described in Section 3.4.2. For 3-dimensional Harmonic tensors ($H > 1$), performance improves as the model uses more harmonics. Note that, as we described in Section 3.4.2, the frequency range in the first harmonic becomes narrower as the number of harmonics H increases ($f_{max} = f_s/2H$). We hypothesize that this is the reason why there is a slight performance drop between $H = 1$ and $H = 2$. However, much larger H might yield worse results. If $H = 10$ for example, the maximum frequency of the first harmonic becomes 800Hz, which means the Harmonic tensor cannot include the harmonic information of higher pitches, i.e., fundamental frequencies higher than 800Hz.

We also tried to determine the role of learnable center frequencies f_c

but we could not find significant differences between learnable and fixed center frequencies. Their performance gaps in three different tasks are all in the range of performance variance. In our experimental setup using quarter tone MIDI scale, there is no observable benefit of using learnable center frequencies f_c.

Finally, we show the role of the bandwidth parameter Q. In this experiment, we used fixed values of α and β with the empirical values [129] and only let Q to be trained. As we mentioned in Section 3.4.2, the Harmonic tensor is more flexible than HCQT since this parameter does not need to be heuristically set. In Table 3.4, the bandwidth parameter Q changes based on task, FFT size, and center frequency interval. This proves that the optimal parameter Q is task- and settings-dependent, thus showing the importance of automatically learning it in a data-driven manner.

3.4.5 Conclusion

In this section, we introduced data-driven Harmonic filters to form a versatile front end for audio representation learning. Experimental results report state-of-the-art performance in automatic music tagging, keyword spotting, and sound event tagging tasks. The output of the proposed front-end keeps locality in time, frequency, and harmonic so that the subsequent back-end can explicitly capture harmonic structures. The proposed front end is flexible since it learns bandwidth parameters in a data-driven fashion. To further scrutinize the representation ability of the proposed model, other complex tasks beyond binary classification should be considered. Analyzing how well this model scales with larger datasets would also be key to better understand the potential of the proposed architecture. Finally, interpretability studies and additional investigation on the learnable parameters of the model may yield valuable insights in terms of how to more optimally apply these Harmonic filters.

3.5 Transformers for music representation

3.5.1 Introduction

Thanks to the recent advances in deep learning, mostly convolutional neural networks (CNNs) [29], the performances of music tagging models have been significantly enhanced by leveraging large-scale data with various deep architectures [115, 112, 57, 108]. However, in this section, we want to highlight two limitations of previous works (instance-level training and less interpretability), and propose new music representation models that alleviate the pointed issues.

Music signals are in the form of sequential data. In this sequence, regarding typical tags, some acoustic characteristics may appear locally (e.g., instruments) while some others may span over the sequence (e.g., mood, genre). This means a successful music tagging model needs to be able to extract both local and global features. Fully convolutional network [115], one of the very early deep learning models for music tagging, was designed to capture both local and global features by increasing the size of the overall receptive fields with max-pooling. More recently, however, it is shown that training with a smaller hop size with shorter audio chunks (i.e. instance-level training) is beneficial for music tagging [104] — see Section 3.3. This approach has been adopted in many CNN-based models [112, 57, 108], where the models are trained with short audio chunks (3 to 5 second long), densely striding max-pooling, and a global pooling layer. To predict music tags of a 3-minute song, for example, the audio is split into multiple short audio chunks, and the model makes predictions on each chunk. Then, the predictions are aggregated through majority vote or global average-/max-pooling. This means that on a track level, the current music tagging models are performing like a bag-of-features model [36] instead of modeling music representation as a sequence.

However, we believe that music is sequential and it composes its high-level semantics based on the relations between individual components in long-term sparse positions, not only based on the local information. On

analogous motivations, Choi et al. adopted convolutional recurrent neural networks (CRNN) [96] and Pons et al. tried to depict deep architectures in two parts: front end and back end [112]. The front end, which is equivalent to the CNN part of CRNN, learns local features. The back end, which corresponds to the RNN part of CRNN, captures the structure of learned local features. Although they reported remarkable results, they are not suitable for modeling the long-term context. To encapsulate long-term context with CNN back end, deep stacks of convolutional layers followed by subsampling layers (mostly max-pooling) are required, which will end up with blurred time resolution. RNN back end with longer sequence inputs suffers from the demand of huge computational power and gradient vanishing/exploding problems [136].

Another limitation of CNN-based models is less interpretability. CNN for MIR are yet less interpretable despite there has been noteworthy previous research to explain the predictions [137, 138, 139]. One possible reason is that spectrogram-based 2D CNN models which have been used in the research learn spectro-temporal characteristics in each layer, while music is a temporal sequence of individual audio events. Highlighted 2D patch in images are very intuitive, while visualization and auralization [137] of spectro-temporal patches in mel spectrograms are still less intuitive for humans to understand the mechanisms behind music representation models.

Self-attention is an attention mechanism that learns a representation by relating different positions in the sequence. It facilitates the model to learn long-term context by relating each pair of positions directly. The transformer [38], which is a sequence model solely based on self-attention, and its variants [30, 140] showed compelling results on extensive NLP tasks. Its versatility is also demonstrated in generative models such as generative adversarial networks (GAN) [141] and auto-regressive models [142, 34]. In particular, the Music Transformer [34] has shown that the transformer could model the long term dependency for musical representations using symbolic data, such as MIDI. And Wave2Midi2Wave [143] expanded the research toward raw audio by adopting the Onsets and Frames [144] to transcribe the raw audio (wave2midi), the Music Trans-

former [34] to generate MIDI notes, and the Wavenet [145] to generate raw audio from the MIDI notes(midi2wave).

Like this, transformer is a powerful representation model which enables long sequence modeling. As we aim at the representation model that performs beyond instance-level training or bag-of-features model, transformer meets the purpose. Another benefit of using transformer is interpretability. Since self-attention mechanism directly learns the relation between different points in the sequence, this can be easily visualized. Inspired by these, we propose to adopt the successful architecture to the back end of music tagging models. By this means, one can expect not only the performance but also the interpretability.

Proposed models in this section consist of CNN front end and transformer back end (Figure 3.7). CNN front end is expected to capture local information: e.g., timbre, pitch, and chord; and the transformer back end to capture more structural information: e.g., rhythmic patterns, melodic contours, and chord progressions; based on the combination of the captured local components. Based on this concept, two different models using transformer are introduced in this section: convolutional neural network with self-attention (CNNSA) [114] and music tagging transformer [113]. Their main differences are temporal resolution and the front end design.

In the following subsection (Section 3.5.2), we introduce the first transformer model for music audio representation learning. Then the upgraded variation, which claims the new state-of-the-art in automatic music tagging, is introduced in Section 3.5.3.

3.5.2 Convolutional neural network with self-attention (CNNSA)

Convolutional neural network with self-attention (CNNSA) [114] is the first attempt to apply transformers in music audio representation learning. In this work, we experimented two different front ends (*Spec* and *Raw*) and compared the proposed transformer back end with conventional CNN back ends.

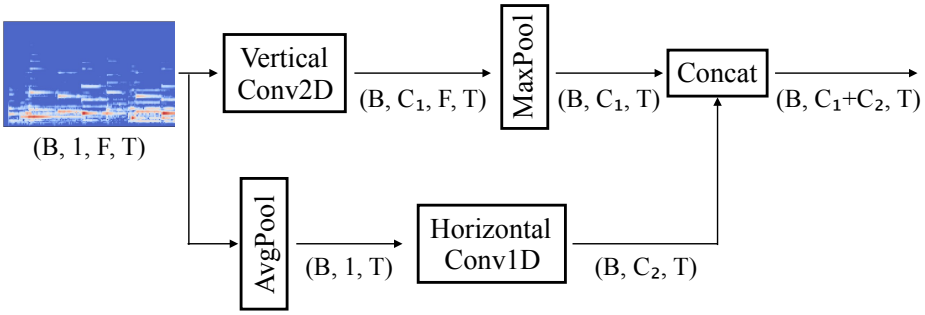


Figure 3.10: *Spec* front-end from MusiCNN. B, C, F, and T stand for batch, channel, frequency, and time dimension.

Front end

Spec front end uses the front end of MusiCNN [112]. As introduced in Section 3.3.3, MusiCNN uses mel spectrogram inputs. Vertically long convolution filters are designed to capture timbre-related features, and horizontally long filters to capture temporal characteristics. After the vertical convolution, extracted feature maps are max-pooled along the frequency axis (see Figure 3.10). By this mean, the appearance of each instrument will be captured while pitch related information to be ignored. Horizontal filters capture temporal energy flux patterns in up to 2.6s sequence. Horizontal filters receive average-pooled (along with frequency axis) spectrograms as their inputs. Since vertical filters have a max-pooling layer after the convolutional layer, and horizontal filters have an average-pooling layer before the convolutional layer, the frequency axis of the tensors can be flattened (Figure 3.10). Flattened two feature maps are concatenated along channels. We call this spectrogram based front end as *Spec* front end. *Spec* front end uses 256 frames (≈4.1s) of spectrogram chunk as its input.

The *Spec* front end is relying on manual design strategies. However, one can expect the data to decide the entire feature design process. From the motivation, we also experimented *Raw* front end. As introduced in Section 3.3.3, sample-level CNN [57, 58] stacks short grain of one di-

Spec		Raw	
Layer	Filter shape	Layer	Filter shape
1	$32 \times 38 \times 1$	1	128×3
1	$32 \times 86 \times 1$	2	128×3
1	$16 \times 38 \times 3$	3	128×3
1	$16 \times 86 \times 3$	4	256×3
1	$8 \times 38 \times 7$	5	256×3
1	$8 \times 86 \times 7$	6	256×3
1	64×33	7	256×3
1	32×65	8	256×3
1	16×129	9	256×3
1	8×165	10	512×3

Table 3.5: Filter shapes of *Spec* front end and *Raw* front/back end. Dimensions of filters are $Channel \times Frequency \times Time$ or $Channel \times Time$.

mensional convolution filters (e.g. 1×3) to model the music sequence. It is an assumption-free model that aims at learning representation from scratch. We call the front end using sample-level CNN as *Raw*. Strictly, there is no clear boundary of front end and back end in the sample-level CNN since it consists of homogeneous 1D convolutional layers. However, to examine our transformer back end, we regarded the first five convolutional layers as a front end since one frame in the feature map after the five layers can include 15.2ms of audio which can be compared with one frame of spectrograms (16ms). Only when we use transformer back end, for the fair comparison, *Raw* front end is followed by one 1×7 convolutional layer since vertical filters of *Spec* front end have capacities of up to 7 frames (112ms). *Raw* front end uses 65,610 samples ($\approx$4.1s) of raw audio inputs. Detailed number of parameters are described in Table 3.5.

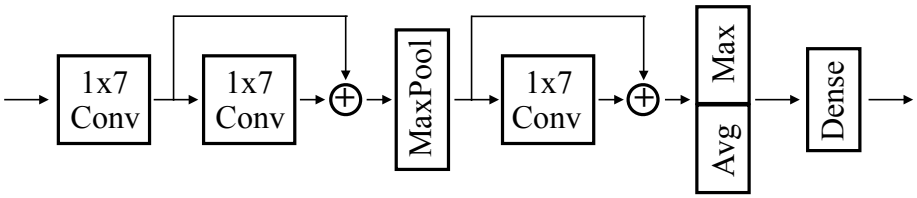


Figure 3.11: CNN_P back end.

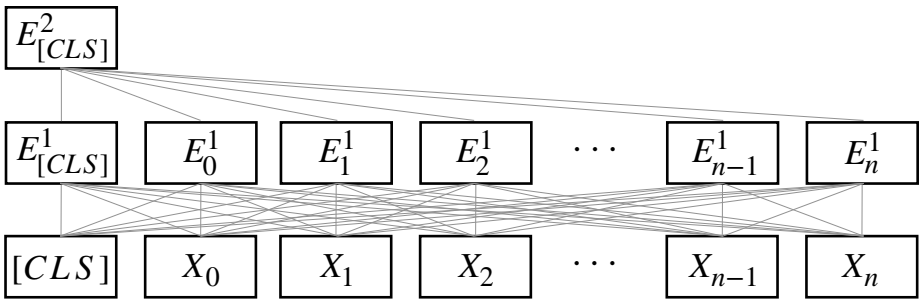


Figure 3.12: *Transformer* back end with two self-attention layers.

Back end

MusiCNN back end uses stacks of 1D CNN with residual connections [79] — see Figure 3.11. Channel size is 512 for each layer. We denote this back end as CNN_P named after Pons et al. [112]. When the model is using *Spec* front end and CNN_P back end (i.e., *Spec_*CNN_P), it is identical to original MusiCNN. On the other hand, as we reviewed before, sample-level CNN does not have a clear boundary of front end and back end. For convenience, we call the latter five layers of sample-level CNN as CNN_L back end named after Lee et al. [57]. In the end, *Raw_*CNN_L model consists of ten 1D convolutional layers as proposed in the original paper [57]. Each layer of both back ends uses batch normalization and ReLU non-linearity.

Our proposed *Transformer* back end is a stack of self-attention layers (Figure 3.12). We applied the self-attention to the feature map that we get

from the front end convolution. If we recall the equation of self-attention in Section 2.4.3, the attention score is

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.4}$$

where d_k is a dimension of keys and Q, K, V are matrices whose shapes are $Sequence \times Embedding$. Suppose a convolution feature map is given after the front end convolution of ***Spec*** or ***Raw*** and let $X \in \mathbb{R}^{T \times CF}$ or $X \in \mathbb{R}^{T \times C}$ denote the feature map, where C is channels, T is time, and F is frequency axis. For simplification, here we only explain with $X \in \mathbb{R}^{T \times C}$ which is a feature map of ***Raw*** front end. In this case, an $1 \times C$ vector of the feature map at each time bin can be regarded as a word embedding. Hence, Q, K, and V of the feature map X can be denoted as:

$$\begin{aligned} Q &= \theta(X) = XW_\theta \in \mathbb{R}^{T \times C} \\ K &= \phi(X) = XW_\phi \in \mathbb{R}^{T \times C} \\ V &= g(X) = XW_g \in \mathbb{R}^{T \times C} \end{aligned} \tag{3.5}$$

where $\theta(\cdot)$, $\phi(\cdot)$, $g(\cdot)$ are learnable transformations.

As the Bidirectional Encoder Representations from Transformers (BERT) [30] achieved remarkable performance in many classificaiton tasks by only using stacked self-attention layers (i.e., transformer encoders), and our task is also to classify not to generate, we only adopted the encoder part of the transformer. As shown in Figure 3.12, our proposed back end uses stacks of self-attention to classify the tags of given sequence X. $[CLS]$ is a special token that includes overall context for the classification. Self-attention that we used is multi-head attention [38].

Optimization

Careful design of learning rate schedule is critical to both of convergence speed and generalization [146, 147]. Adam [109], an adaptive optimization method, achieves fast convergence but it is generally known to impede the generalization of models [148, 149]. Instead of using conventional stochastic gradient descent (SGD) or Adam, we propose an opti-

mization technique inspired by the Switches from Adam to SGD (SWATS) [148].

We first optimize the network using Adam [109] with learning rate $1e{-}4$, beta1 0.9, and beta2 0.999. After 60 epochs, we reload the model which achieved the best validation ROC-AUC during the 60 epochs, and switch the optimizer to SGD with momentum 0.9 and nesterov momentum. We drop the learning rate by 10% at the epoch 80 and 100. In Section 3.5.2, we show that our proposed mixed optimization scheme improves the generalization capacity than an SGD with manual learning rate scheduling. Note that our proposed method loads the best model weights during the training while SWATS [148] switches optimizer without changing the weights.

Dataset

We used MagnaTagATune dataset (MTAT) [54] and the million song dataset (MSD) [43] for our experiments. Details of the datasets are described in Section 3.2.1. Dataset splits are identical to Section 3.3.2 [104].

We investigate two different types of input: raw audio and log mel-spectrogram. For the comparable research, we decided to use 16kHz sampling rate for both inputs. Essentia library [150] was used to load and downsample the audio. To get the log mel spectrograms, hanning window of 512 samples with 50% overlap has been used and the number of mel bins was set to 96. Librosa library [151] was used for this step. We did not normalize the dataset. Instead, CNN_P has batch normalization in the first layer.

Results

We report ROC-AUC and PR-AUC to assess different models. Since we are using user-generated tags (MTAT and MSD), there is popularity biased skewness in their distributions. Although we are using ROC-AUC to choose the best model, it's not always the best in both metrics — see Table 3.7.

		MTAT		MSD	
Front end	Back end	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
Raw [57]	CNN_L [57]	90.62	44.20	88.42*	-
Raw [57]	*Transformer* (Ours)	90.66	44.21	88.07	29.90
Spec [112]	CNN_P [112]	90.89	45.03	88.75*	31.24*
Spec [112]	*Transformer* (Ours)	90.80	44.39	88.14	30.47

Table 3.6: Comparison of state-of-art music tagging models on MTAT and MSD. The results marked with (*) on top are reported values from the reference papers.

# heads	# layers	ROC-AUC	PR-AUC
1	2	87.73	36.93
2	2	89.40	41.20
3	2	90.23	43.23
4	2	90.40	43.89
5	2	90.60	43.91
6	2	90.61	44.39
7	2	90.74	**44.43**
8	2	**90.80**	44.39
8	1	90.54	44.12
8	2	**90.80**	**44.39**
8	3	90.19	43.22

Table 3.7: Impact of the number of attention heads and layers on MTAT.

Input length	# layers	ROC-AUC	PR-AUC
256	2	90.80	44.39
1024	2	89.62	41.61
1024	3	89.85	**42.25**
1024	4	**89.86**	41.84

Table 3.8: ROC-AUC and PR-AUC results on MTAT using proposed *Spec_Transformer* models with longer input sequence.

Table 3.6 shows ROC-AUC and PR-AUC of the baseline models and our proposed models. Each value in the table is the average of three different runs. As shown in the table, our proposed *Transformer* back end reports competitive results for both datasets.

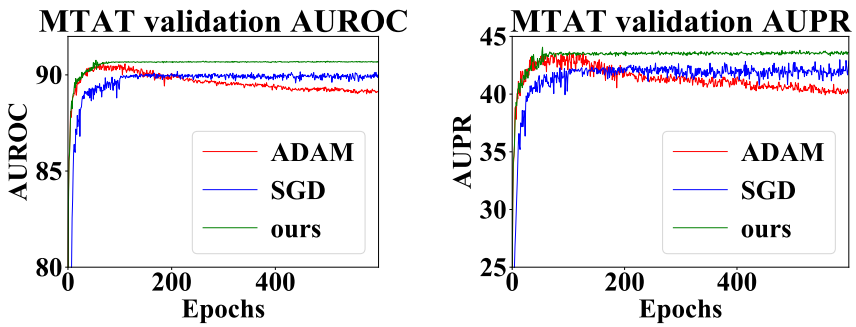


Figure 3.13: Comparison of optimizers: Adam, SGD, and our proposed method.

Attention Parameters. Choosing an appropriate number of attention layers and heads can be crucial for designing better models. As shown in Table 3.7, attention layers more than 2 did not show significant improvement and 8 attention heads reported the best performance. Hence, we fixed the number of attention layers and attention heads in our experiments as 2 and 8, respectively. Note that this setup is optimized for ≈4.1s inputs.

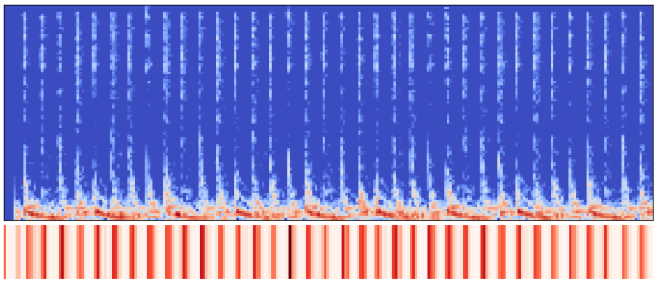
Optimization. As we depicted before, we used our novel optimization method. By adopting Adam [109] in the beginning, we expected faster convergence than SGD. As shown in Figure 3.13, Adam and our optimization method show a steeper learning curve than SGD. However, ROC-AUC and PR-AUC of Adam go down after around 100 epochs, which means it failed to generalize the model. Since we switch our model to SGD at 60 epochs, it shows more stable learning curve than Adam only. Although this switch point is an arbitrary point, our optimization method can generalize the model well because we load the best model during the training when we switch the optimizer or learning rate — we used ROC-AUC to choose the best model.
Longer Sequence. In our main experiment, we only used relatively short audio chunks (≈4.1s) as our input for the fair comparison — sample-level CNN used short chunks. However, transformer is known to be efficient to model long-term sequence. We experimented the *Spec_Transformer* model for MTAT using 1024 samples (≈16.4s) and we could see slightly lower but comparable results — see Table 3.8. More stacks of self-attention layers were required to model longer sequence.

Visualization

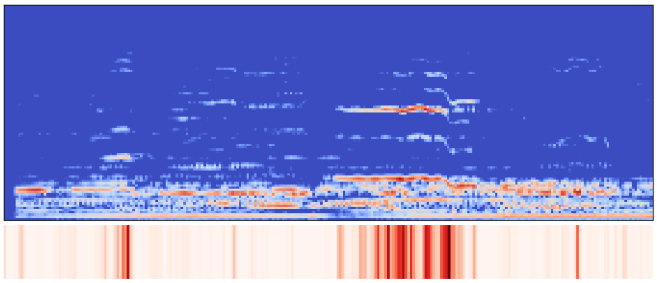
To interpret the proposed model, we provide two different visualization: attention heat map and tag-wise contribution heat map. While attention heat map shows where the trained model pays more attention, tag-wise contribution heat map highlights which part of the input spectrogram is more relevant to predict the given tag.
Attention Heat Map. To understand the behavior of the model, it is important to know which part of the audio the machine pays more attention to. To this end, we summed up attention scores from each attention head and visualized. Attention score A of a single attention head can be described as:

$$A = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right). \tag{3.6}$$

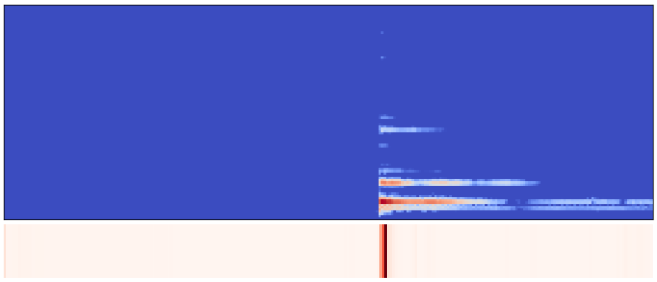
Figure 3.14 shows log mel-spectrograms and according attention heat maps. For simplification, we only visualized the attention heat map of



(a) Tag - Beats

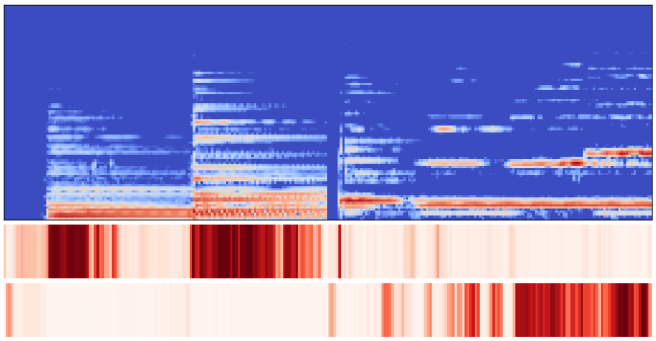


(b) Tag - Female

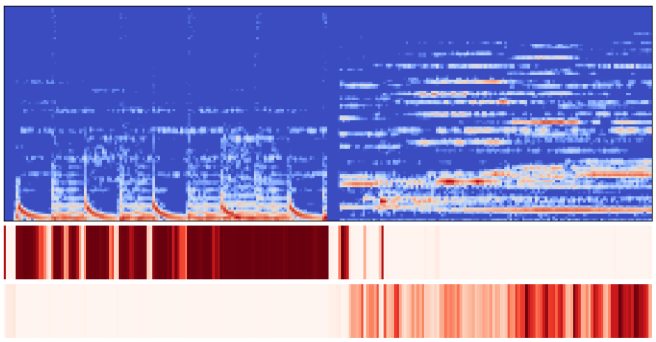


(c) Tag - Quiet

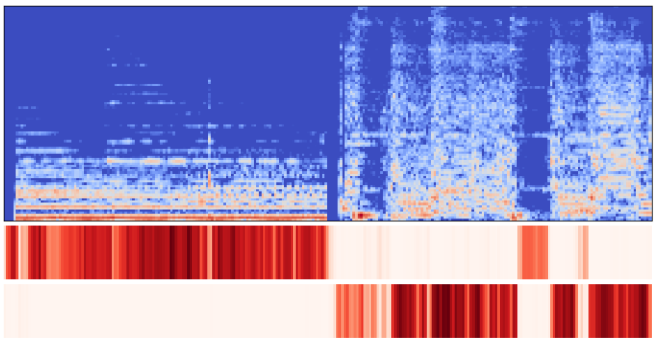
Figure 3.14: Attention heat maps.



(a) Piano + Flute



(b) Techno + Classic



(c) Quiet + Loud

Figure 3.15: Tag-wise contribution heat maps on concatenated spectrograms. From the top, concatenated spectrograms, contribution heat maps to the first tags (Piano, Techno, and Quiet, respectively), and contribution heat maps to the second tags (Flute, Classic, and Loud, respectively).

the last attention layer. As we can see in Figure 3.14a and Figure 3.14b, the model pays more attention to relevant parts of spectrograms. However, we discovered one interesting thing which is: the model always pays attention to the parts with audio events. For example, in Figure 3.14c, the model pays attention to the loud part of the audio although the given spectrogram was classified as "quiet". We could also observe this behavior from negative tags such as "no vocal", "no vocals", and "no voice". One possible reason is that the model pays attention to the more informative part of the spectrogram. Indeed, negative tags report relatively worse ROC-AUC (≈ 0.7) than other tags (≈ 0.9). Although attention heat maps can pinpoint where the machine pays attention for the decision, they cannot provide reasons for the classification or tagging.

Tag-wise Contribution Heat Map. Understanding which part of the audio is more relevant to each tag is also important to interpret the model. We manually changed the attention score of the last attention layer. For each time step, we manipulated the attention score as 1 and set other parts as 0 so that we can see the contribution of each time bin to each tag. This tag-wise contribution heat map is inspired by the manual attention weight adjustment proposed in [152]. To compare the different contribution of different audio, we concatenated two spectrograms and fed them through the network. For instance, Figure 3.15a is a concatenated spectrogram of piano (left half) and flute (right half). The first row heat map highlights the contribution of each time bin to the "piano" and the second row is for "flute". We repeated this for genre (Figure 3.15b) and mood (Figure 3.15c). As shown in Figure 3.15c, the tag-wise contribution heat map can provide more information about tag specific part of the audio, which was not able to be observed from the attention heat map (Figure 3.14c).

Conclusion

In CNNSA research, we proposed a novel deep sequence model for music tagging using transformer which can facilitate better interpretability. The proposed model consists of CNN front end and transformer back end. Experiments on MTAT dataset and MSD reported competitive results and

we could demonstrate the interpretability of the model by visualizing attention heat maps and tag-wise contribution heat maps. By leveraging the acquired interpretation, one can obtain better intuition for the model design.

3.5.3 Music tagging transformer

In this subsection, we introduce an advanced music representation model using transformer: music tagging transformer [113].

Front end

Two of the main conclusions of Section 3.3 recommend (*i*) using mel spectrogram inputs, and (*ii*) using the most granular 2D filters (i.e., 3×3 convolution) instead of manual design choices. That means, a simple 2D CNN with mel spectrogram inputs, which is prevalent and sometimes referred to as *vgg-ish* model, is still outperforming the other music tagging models. We follow the suggestions – we use 3×3 convolution filters with residual connections [79] on mel spectrogram inputs. Table 3.9 outlines our 3-layered CNN front end where B is the batch size, C is the number of convolution channels, F is the number of mel bins, T is the number of frames, and C' is the number of attention channels of Transformer. This CNN front end (*i*) helps the model to capture local representations and (*ii*) reduces the time resolution of the input so that it is feasible to train the following back end.

At the end of the CNN, the second and the third dimensions are reshaped into a single dimension. This flattening is motivated by Vision Transformer (ViT) [87] which reshapes a 2D image patch into a one-dimensional array. As a result, the output of the CNN is a sequence of short-chunk audio features where a chunk corresponds to approximately 0.1 second. It is input to the back end transformer. This is in contrast to the CNNSA [114] that used the frequency-axis max-pooling at the end of its front end. In other words, in Music Tagging Transformer, the attention layers are given more detailed spectral information.

layer	output shape
Input	$B \times 1 \times F \times T$
Conv (3×3)	$B \times C \times F \times T$
MaxPool (2×2)	$B \times C \times F/2 \times T/2$
Conv (3×3)	$B \times C \times F/2 \times T/2$
MaxPool (2×2)	$B \times C \times F/4 \times T/4$
Conv (3×3)	$B \times C \times F/4 \times T/4$
MaxPool (2×1)	$B \times C \times F/8 \times T/4$
Reshape	$B \times (C \cdot F/8) \times T/4$
Fully-connected	$B \times C' \times T/4$

Table 3.9: Front end CNN of Music Tagging Transformer.

Back end

Our back end Transformer architecture is nearly identical to the previous works [30, 114] except for the number of parameters and input lengths. After a hyperparameter search, we chose 4 layers, 256 attention dimensions, and 8 attention heads. At the input stage, positional embedding [30] is applied and a special token embedding $E_{[CLS]}$ is inserted so that the Transformer can perform sequence classification as a downstream task.

Data Augmentation

Data augmentation is a key to success in generalizable representation learning. In our experiments, we take advantage of *Audio Augmentations* library [153] which is easily integrated to PyTorch data pipeline. The applied data augmentation methods are as follows:

- **Polarity inversion**.
- **Additive noise** by $k_{snr} \in \{0.3, 0.5\}$.
- **Random gain** by $\mathcal{A} \in \{-20, -1\}$ dB.

- **High-pass filter** by $f_H \in \{2200, 4000\}$ Hz.
- **Low-pass filter** by $f_L \in \{200, 1200\}$ Hz.
- **Delay** by $t \in \{200, 500\}$ ms.
- **Pitch shift** by $n \in \{-7, 7\}$ semitones.
- **Reverb** by room size $s \in \{0, 100\}$.

Each augmentation method is activated independently with a probability $p \in \{0.3, 0.7\}$.

Dataset

For a consistent comparison, we used the conventional data split of MSD that was used in previous works [115, 112, 57, 108, 104]. All the audio signals are preprocessed to 22,050 Hz sample rate and converted to short-time Fourier transform representations with a 1024-point FFT and 50%-overlapping Hann window. Finally, we convert them to log mel spectrograms with 128 mel bins.

We also suggest a new split to alleviate some known problems of the conventional split. There are two problems – First, since the MSD music tags are collected from users, some of them are very noisy, and that may lead to noisy (and incorrect) evaluation [105]. Second, a strict split of music items requires taking the artist information into consideration since often, songs and labels from the same artist heavily resemble each other. However, the conventional split was done without such consideration, having caused unintended information leakage between the training and evaluation sets. Ultimately, this would cause an overly optimistic evaluation. As a solution, we use manually cleaned data from a previous work [91] and take the top 50 tags. We also propose a new split of MSD that does not share any artist among training/validation/test sets and is extended to more tracks. We name this 'CALS split' (cleaned and artist-level stratified split). CALS split consists of 233k labeled tracks and 516k unlabeled tracks.

Performance with the conventional split

The model is optimized using Adam [109] with a learning rate of 0.0001. The best model is selected based on the binary cross entropy loss of the validation set and early stopping is applied when the validation loss does not improve for 20 epochs.

Table 3.10 summarizes the performance of previous systems and the proposed model using the conventional split. The ROC-AUC and PR-AUC of the many previous models have been under 0.89 and 0.33, respectively. Our model, Music Tagging Transformer, outperforms the previous state-of-the-art models, harmonic CNN and short-chunk ResNet. The improvement, especially on PR-AUC, is non-trivial and even larger with data augmentation.

The front end of our Music Tagging Transformer takes a sequence of chunks, where each of which represents a very short duration of the signal ($\approx$0.1 second) [104]. Because 0.1 second would be too short to represent musical characteristics alone, we interpret that the experimental results would mean our transformer back end plays a role of sequential feature extractor beyond simple bag-of-feature aggregation. This may be an important aspect of the proposed model since sequential modeling is what the self-attention mechanism is the best suit.

The data augmentation we adopted contributes to improvements of 0.0056 ROC-AUC and 0.0119 PR-AUC. These are bigger than many of the improvements we have seen between different architecture choices. This emphasizes that data augmentation should be considered when developing a music tagging model.

Performance with the CALS split

For a deeper and more accurate analysis of the proposed model, we also use the proposed CALS split. Table 3.11 presents the experimental results of short-chunk ResNets (previous state-of-the-art architecture) and Music Tagging Transformers. The Music Tagging Transformer consistently outperforms ResNet models in both conventional and CALS splits.

Models	ROC-AUC	PR-AUC
FCN [115]	0.8742	0.2963
Musicnn [112]	0.8788	0.3036
Sample-level [57]	0.8789	0.2959
Sample-level+SE [58]	0.8838	0.3109
CRNN [96]	0.8460	0.2330
CNNSA [114]	0.8810	0.3103
Harmonic CNN [108]	0.8898	0.3298
Short-chunk CNN [104]	0.8883	0.3251
Short-chunk ResNet [104]	0.8898	0.3280
Transformer (proposed)§	**0.8916**	**0.3358**
Transformer (proposed) + DA†	**0.8972**	**0.3479**

Table 3.10: Performance comparison using the conventional MSD split for top-50 music tagging. The § and † marks mean they are based on the identical model architecture and training strategy; compared to the same, marked models in Table 3.11, only the dataset split is different.

For both of the models, it also summarizes the results of vanilla models and with data augmentation (DA). Note that the two bottom rows of Table 3.10 correspond to the 3rd and 4th rows of Table 3.11 as marked with § and †. For both short-chunk ResNet [104] and the Music Tagging Transformer, we observe constant improvements when data augmentation is applied.

More hyperparameter search

We further present the experiment results with various model configurations. First, we trained our Music Tagging Transformer and short-chunk ResNet with varying input lengths to assess our proposed model's ability to handle long sequences. As shown in Figure 3.16, on both of the metrics, short-chunk ResNet shows a noticeable performance degradation as the audio input gets longer. This shows that the global max pooling in the

Models	#param	ROC-AUC	PR-AUC
ResNet [104]	13.5m	0.9098	0.3525
ResNet+DA	13.5m	0.9141	0.3705
Transformer[§]	4.6m	**0.9188**	**0.3775**
Transformer+DA[†]	4.6m	**0.9191**	**0.3845**

Table 3.11: Performance comparison using the CALS MSD split for music tagging.

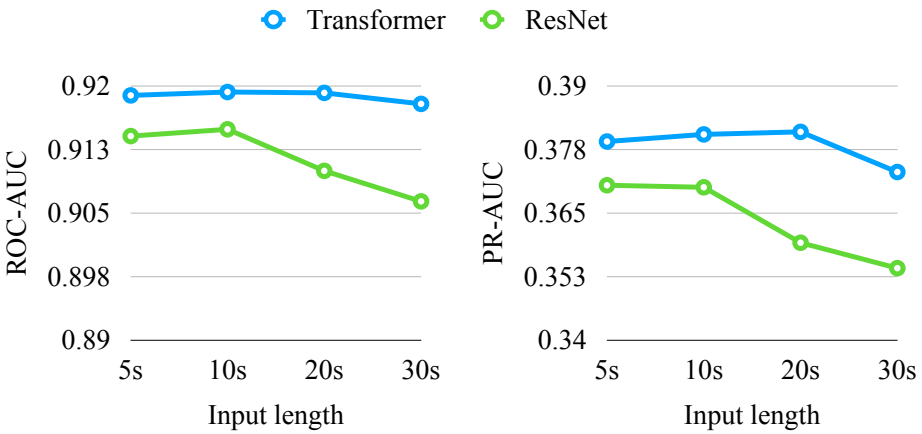


Figure 3.16: Performance with different input lengths.

short chunk ResNet is not perfectly suitable for a long signal. Meanwhile, the Music Tagging Transformer shows consistent performances in general. An exception is when the input is 30-second long. We suspect the performance drop of the Music Tagging Transformer happens because the model cannot take advantage of random cropping data augmentation effect since the 30-second is the full length of the MSD previews.

Second, we investigate different Transformer parameters to figure out the best performing setup. As summarized in Table 3.12, transformer achieved the best performance with attention channels (width) at 128 and 256, and their depth of 4 and 8 layers. However, these optimal parameters

width	depth	ROC-AUC	PR-AUC
32	4	0.9118	0.3528
64	4	0.9178	0.3754
128	4	**0.9194**	0.3776
256	4	0.9191	**0.3845**
512	4	0.9177	0.3788
768	4	0.9162	0.3707
1024	4	0.9174	0.3736
256	1	0.9180	0.3736
256	2	0.9193	0.3805
256	4	**0.9199**	0.3814
256	8	0.9181	**0.3826**
256	12	0.9165	0.3780
256	16	0.9169	0.3785

Table 3.12: The performance of Music Tagging Transformer with varying width and depth of the attention layers.

are dataset-dependent; as generally observed, a larger network structure would perform better if a larger amount of training data is provided.

Conclusion

In this subsection, we proposed a new architecture, Music Tagging Transformer, and improved its tagging performance with data augmentation. Experimental results showed that the proposed architecture outperforms the previous state-of-the-art models in supervised music tagging using the MSD [43]. We also provided an analysis result that shows Music Tagging Transformer can handle long audio inputs better than the previous CNN architectures do.

In future work, our transformer can be further utilized in various MIR tasks. Since Transformer can perform both sequence-level and token-level classification, it can be used in not only music tagging but also tasks such

as beat detection and melody extraction. Finally, by combining the multiple MIR tasks in a multi-task learning scheme, Transformer can be trained as a general purpose music representation learning model.

3.6 Summary

This chapter investigated various automatic music tagging architectures as a proxy of music representation learning. Section 3.3 introduced conventional approaches for music tagging and evaluated the models under the same experimental setup. Before this research, it was difficult to compare different models, but now we know the pros and cons of various representation models. The main conclusions are (*i*) manual design in the front end can be helpful when a dataset is small, (*ii*) but assumption-free models outperform as the size of the dataset grows, (*iii*) however, minimum preprocessing (i.e., mel spectrogram) is yet required to achieve the best performance, (*iv*) instance-level training is more powerful than song-level training. Another contribution of this work is a reproducible code [7]. All different models are implemented in PyTorch so that other researchers can use them efficiently.

Based on the knowledge that assumption-free 3×3 filters are powerful in representation learning, we introduced data-driven harmonic filters [108] in Section 3.4 to take advantage of both domain knowledge and data-driven approaches. The proposed approach reported advanced performance not only in music tagging but also in keyword spotting and acoustic event detection. Also, it showed better generalizability than other CNN-based models when the input audio is transformed with unseen types of deformation during the training.

Finally, we introduced transformer [38, 30] to music representation learning. The proposed model consists of a CNN front end that learns local acoustic characteristics and a transformer back end that summarizes the sequence of the local features. The music tagging transformer claimed the new state-of-the-art in music tagging, and it successfully manages

[7] https://github.com/minzwon/sota-music-tagging-models.git

long-sequence modeling. Also, the transformer back end provides better temporal interpretability of the model's behavior.

Based on this chapter's learned knowledge about deep architecture design, we further improve their performances in the next chapter (Chapter 4) by switching the training scheme. Also, some of the pretrained models are utilized in Chapter 5 to facilitate multimodal music representation learning.

Chapter 4

REPRESENTATION LEARNING AT SCALE

4.1 Introduction

In previous chapter, we explored various architecture designs to improve the performance of music classification models. All introduced models are experimented using labeled data, such as MTAT [54], MSD [43], and MTG-Jamendo dataset [40], under a supervised learning scheme. However, as we discussed in Section 2.3, collecting music labels for supervised learning is challenging, hence expensive. To overcome the issue of limited labels, and step further with more scalable research, we need training schemes beyond supervised learning. This chapter introduces transfer learning [63] and semi-supervised learning [76] to automatic music tagging, so that we can utilize external labeled data and unlabeled data to enhance the model's generalizability.

This chapter is organized as follows. Section 4.2 introduces a transfer learning method for music genre classification. It depicts the winning submission model at Learning to Recognize Musical Genre from Audio challenge in *The Web Conference 2018*. Then Section 4.3 experiments a successful semi-supervised learning scheme: noisy student training [76]. It discusses how can we utilize abundant unlabeled data effectively in au-

tomatic music tagging. Finally, Section 4.4 summarizes the introduced approaches and discusses future directions.

This chapter includes the following works:

- Jaehun Kim, Minz Won, Xavier Serra, and Cynthia CS Liem, Transfer Learning of Artist Group Factors to Musical Genre Classification, The Web conference (WWW) 2018 challenge. [1]

- Minz Won, Keunwoo Choi, and Xavier Serra, Semi-supervised Music Tagging Transformer, The International Society for Music Information Retrieval (ISMIR) 2021.

4.2 Transfer learning of artist group factors

4.2.1 Challenge

Learning to Recognize Musical Genre from Audio is a challenge track of *The Web Conference 2018*. The challenge is to predict correct musical genres of given audio using the Free Music Archive (FMA) dataset [45]. Since the test set was not known, our main objective is to build a generalizable machine for music genre classification.

Before we start training models, we first carefully reviewed the dataset. FMA dataset [45] is a modern, large-scale dataset that contains full-tracks, instead of short preview clips. Genre labels are chosen by the artists from a pre-defined genre hierarchy. The subset we used in this challenge includes 25,000 tracks from 5,152 unique albums. For 5,028 out of these 5,152 albums, genre tags have been labeled at the album level. All tracks in an album can share a homogeneous genre but this is not always true. Indeed we could discover multiple misannotations in the dataset. As a result, our challenge is to build a generalizable machine for music genre classification using noisy training data.

[1] I am the second author of this paper. Section 4.2 introduces the overall concept of the proposed approach but the main contributor of this work is the first author. My main contributions in this paper are model implementation and experiments.

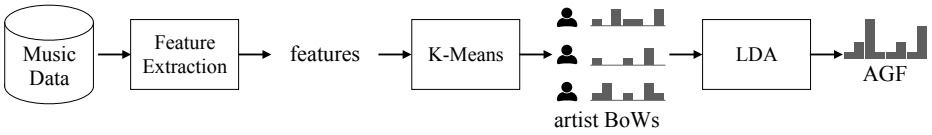


Figure 4.1: Artist group factor extraction pipeline.

4.2.2 Proposed approach

There can be multiple possible solutions for handling the challenge. One can manually clean the training data, or regularize the prediction entropy [70, 71] to alleviate the impact of noisy labels. In this challenge, we adopted a transfer learning scheme by defining the source task with more objective and consistent data, i.e., artist labels. For most tracks, we have artist metadata, and this information is objective. A previous work [39] has shown that the learned representation from artist labels can be effectively transferred to solve other music-related downstream tasks. However, sometimes there are only a few tracks for each artist and this makes the training data more sparse. In this work, we group acoustically similar artists to generate Artist Group Factors (AGF) and train our model to predict the AGF. Finally, we transfer the learned representation to solve our original task: genre classification.

Artist group factor (AGF)

When we have labels, such as genres and subgenres, we can construct a Bag-of-Word (BoW) artist-level feature vector by counting the labels from the songs of the artist. Also, we can build the BoW vector using acoustic features. For example, we can extract MFCC from the entire data and cluster the frame-level features through K-means clustering [154]. Then we can count the latent MFCC 'terms' belonging to each artist to create the artist BoW feature vector. Once artist-level BoW feature vectors are prepared, we apply more sophisticated topic modeling algorithm (Latent Dirichlet Allocation (LDA) [155]) to generate the artist group factor (AGF). The AGF extraction pipeline is illustrated in Figure 4.1.

Table 4.1: Details of Learning Targets

id	Category	Source task	Clustering	Dimension
g	Main	Genre	N / A	16
m	AGF	MFCC	K-means	25
d		dMFCC		25
e		Essentia [150]		4374
s		Subgenre	N / A	150

As shown in Tabel 4.1, 16-genre labels and 150-subgenre labels are used together with MFCC, dMFCC, and Essentia features [150]. Essentia music feature extractor extracts descriptors ranging from low-level features, such as statistics of spectral characteristics, to high-level features, including danceability [156] or semantic features learned from the data.

Model architecture

We used 7-layer convolutional neural network (CNN) to predict the AGFs and fine-tuned it for genre classification. It takes mel spectrogram inputs with 1 second of audio. Note that we participated in the challenge in 2018 which was before the holistic model comparison introduced in Section 3.3. The detailed model architecture of our deep convolutional neural network is described in Table 4.2. The size of the output layer is 16 when it is trained to optimize genre classification, directly. Otherwise, it is targeting 40-dimensional AGF.

Transfer learning

After training source task models using the AGFs, we fine-tuned the pretrained models to solve the target task: genre classification. An MLP with one hidden layer is added to the penultimate layer of the pretrained model. We also experimented ensemble model by concatenating the embeddings from different AGF models. As shown in Figure 4.2, the embeddings of the penultimate layers are concatenated when we train an ensemble

Table 4.2: Proposed CNN structure

Layers	Output shape
Input layer	$128 \times 43 \times 1$
Conv 5×5, ELU	$128 \times 43 \times 1$
MaxPooling 2×1	$64 \times 43 \times 16$
Conv 3×3, BN, ELU	$64 \times 43 \times 32$
MaxPooling 2×2	$32 \times 21 \times 32$
Dropout (0.1)	$32 \times 21 \times 32$
Conv 3×3, ELU	$32 \times 21 \times 64$
MaxPooling 2×2	$16 \times 10 \times 64$
Conv 3×3, BN, ELU	$16 \times 10 \times 64$
MaxPooling 2×2	$8 \times 5 \times 64$
Dropout (0.1)	$8 \times 5 \times 64$
Conv 3×3, ELU	$8 \times 5 \times 128$
MaxPooling 2×2	$4 \times 2 \times 128$
Conv 3×3, ELU	$4 \times 2 \times 256$
Conv 1×1, BN, ELU	$4 \times 2 \times 256$
GlobalAveragePooling, BN	256
Dense, BN, ELU	256
Dropout (0.5)	256
Output layer 16 or 40	16 or 40

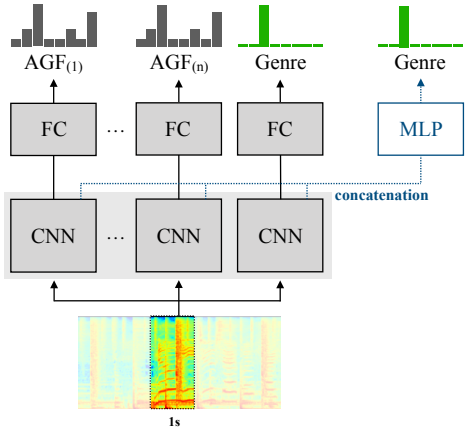


Figure 4.2: Illustration for the transfer learning scenario. Dotted lines indicate the setup for the multilayer perceptron for performing final genre classification.

model. Furthermore, we wanted to see if multi-task learning is beneficial when a model is optimized to predict multiple AGFs at the same time.

4.2.3 Results

As shown in Table 4.3, there was no significant performance gap between single-task and multi-task learning. However, we could observe the best performance when we transfer and ensemble all the AGF models no matter if the model is transferred from single-task or multi-task learning. We conclude that various AGF source tasks bring more generalizable representation which can improve the target task performance. Especially, when the given labels are noisy, one can take adventage of more objective features to generate AGFs and predict them as a pretext task. Through this approach, our model could win the Learning to Recognize Musical Genre from Audio challenge at *The Web Conference 2018*.

Table 4.3: The performance of various combinations of AGFs and the top-level main genre target as a feature learning task.

	STN		MTN	
	LogLoss	F1	LogLoss	F1
g	0.8891	0.5963		
m	1.1812	0.3581		
d	1.0987	0.3967	N/A	N/A
e	1.2542	0.3437		
s	0.9404	0.5218		
gs	0.8606	0.6114	0.8578	0.6190
ge	0.8811	0.5953	0.8792	0.5996
gd	0.8845	0.5898	0.8803	0.5955
gm	0.8874	0.5957	0.8813	0.6037
se	0.9124	0.5537	0.9079	0.5502
sd	0.9191	0.5601	0.9146	0.5412
sm	0.9260	0.5581	0.9283	0.5458
ed	1.0557	0.4433	1.0422	0.4399
em	1.1186	0.4244	1.1060	0.4376
dm	1.0583	0.4373	1.0704	0.4280
gse	0.8361	0.6255	0.8335	0.6277
gsd	0.8579	0.6280	0.8519	0.6150
gsm	0.8486	0.6289	0.8541	0.6153
ged	0.8528	0.6051	0.8601	0.6067
gem	0.8645	0.5988	0.8701	0.6056
gdm	0.8773	0.5985	0.8845	0.5941
sed	0.8965	0.5818	0.8867	0.5640
sem	0.9104	0.5834	0.8889	0.5668
sdm	0.9211	0.5629	0.9109	0.5572
edm	1.0359	0.4879	1.0365	0.4675
gsed	0.8211	0.6343	0.8132	0.6328
gsem	0.8264	0.6352	0.8172	0.6284
gsdm	0.8407	0.6379	0.8288	0.6170
gedm	0.8466	0.6053	0.8450	0.6152
sedm	0.8906	0.5856	0.8875	0.5870
gsedm	**0.7894**	**0.6599**	**0.7727**	**0.6571**

4.3 Semi-supervised music tagging

4.3.1 Introduction

One limitation of the current music tagging research is a limited amount of labeled data. Modern deep learning models are data-hungry. However, manually labeling music with tags is time-consuming and requires domain expertise. In pursuit of large-scale research, the million song dataset (MSD) [43], which literally includes a million songs in it, became popular in music tagging research. Among the million songs, however, only about 24% are labeled with at least one of the top-50 music tags. Most of the previous music tagging research has only utilized the labeled data while discarding 76% of the songs in the dataset. This type of setup (i.e., a small labeled dataset along with a large unlabeled dataset) is not limited to the MSD but can be found often in the real world regardless of the domain. To leverage the unlabeled data, self-supervised [81, 77, 78, 97, 80] and semi-supervised [157, 76, 158] learning have been actively explored in computer vision and natural language processing (Section 2.3).

In this section, we explore a successful semi-superivsed learning approach from computer vision: noisy student training [76]. Through this approach, we can include both labeled and unlabeled data in our training process. To the best of our knowledge, this is the first attempt to utilize the entire audio of MSD [43].

4.3.2 Semi-supervised Learning

With the advances of scalable hardware and training algorithms, the demand for labeled data has outpaced the progress of the size of datasets in many fields. As a solution, researchers started to develop methods that can take advantage of unlabeled data. Self-supervised [81, 77, 78, 97] and semi-supervised learning [157, 76, 158] aim at leveraging the abundant unlabeled data and have shown strong performances in various domains.

In many self- and semi-supervised learning approaches, the models are trained to return noise-invariant predictions [78, 97, 76], i.e. consis-

tency training (details are introduced in Section 2.3.2). This 'noise' is usually realized in a form of data augmentation. In detail, with a self-supervised learning scheme, models are trained to optimize the agreement between different views of the same input [78, 97].

On the contrary, semi-supervised learning takes advantage of *both* existing labeled data and unlabeled data. One effective way of handling the data is to formalize the problem as teacher-student learning [159, 76, 160]. In teacher-student learning, a teacher model is first trained with labeled data in a supervised scheme and then, a student model is trained to mimic the teacher's behavior by predicting pseudo-labels [71] that are generated by the teacher model. The teacher-student training has been actively explored with the purpose of domain adaptation [159], knowledge distillation [161], and knowledge expansion [76]. Especially, noisy student training [76] successfully takes advantage of the teacher-student training with the aforementioned noise invariance.

4.3.3 Noisy Student Training

To leverage unlabeled data, we investigate noisy student training [76], a successful semi-supervised learning approach. Table 4.4 provides an overview of noisy student training with pseudocode. First, we train a teacher model $\mathcal{T}$ with a conventional supervised learning approach (line 1–6). Then, we train a student model $\mathcal{S}$ with two types of losses. The first loss, l_1, is coming from the typical supervised approach with labeled data (line 10–11) as done for the teacher model. The other loss, l_2, is from unlabeled inputs and the corresponding pseudo-labels provided by the teacher model (line 12–15). In order to make the student model perform beyond mimicking the teacher model, data augmentation is applied (line 13). Both hard (binaries) and soft labels (logits) can be used for the pseudo-labels [76] and we use soft labels in our work. If the training is successful, the trained student model would outperform the teacher model. Furthermore, the whole training process can be done iteratively by using the student as a new teacher model and training another student model to obtain an even better performing model. For a stronger teacher

model, we used data augmentation in our supervised learning pipeline as well (line 1–6 and 10–11). As a result, the only pipeline without data augmentation is pseudo-label generation (line 12).

The size of the student model can be identical or larger than the teacher model. In this case, one can interpret the training process as *knowledge expansion* [76], meaning the knowledge in the teacher model is upgraded in the student model. One can also design the student model to be smaller than the teacher model, making the process *knowledge distillation* [162]. Knowledge expansion and knowledge distillation are complementary; depending on the use-case, one could pursue either performance or efficiency. We investigate both directions in this research.

4.3.4 Dataset

We use the MSD with CALS split which was introduced in Section 3.5.3. It includes 233k labeled tracks with top-50 tags and 516k unlabeled tracks. When noisy student training is used, it is common to have an unlabeled set that is significantly bigger than the labeled set. For example, in computer vision, 81 million unlabeled items were used along with 1.2 million labeled items [76], making the ratio of the semi-supervised set to be 67.5 $(81/1.2)$. However, with 233k labeled items and 516k unlabeled items, our ratio is only around 2.3. This might be a factor that limits us from fully exploring the potential advantage of semi-supervised learning as we will discuss in Section 4.3.6.

4.3.5 Models

Based on our previous research (Chapter 3), we use two different representation models for our experiment. One is simple but powerful short-chunk ResNet [104], and another is the new state-of-the-art Music Tagging Transformer [113]. We apply noisy student training to both models to see if the training scheme is generalizable and model-agnostic.

Noisy Student Training

Input labeled data X, labels Y, unlabeled data Z
Models teacher model $\mathcal{T}$, student model $\mathcal{S}$
Functions loss function $\mathcal{L}$, data augmentation $\mathcal{A}$,
back propagation $\mathcal{B}$
Train

```
for x ∈ X, y ∈ Y
    do
    p ⟵ 𝒯(x)            // predict
    l ⟵ ℒ(p, y)         // get loss
    𝒯 ⟵ ℬ(𝒯, l)         // update teacher model
    end do
end for
for x ∈ X, y ∈ Y, z ∈ Z
    do
    p₁ ⟵ 𝒮(x)           // predict
    l₁ ⟵ ℒ(p₁, y)       // get supervised loss
    ψ ⟵ 𝒯(z)            // generate pseudo-label
    ẑ ⟵ 𝒜(z)            // data augmentation
    p₂ ⟵ 𝒮(ẑ)           // predict
    l₂ ⟵ ℒ(p₂, ψ)       // get semi-supervised loss
    𝒮 ⟵ ℬ(𝒮, l₁ + l₂)   // update student model
    end do
end for
```

Table 4.4: Pseudocode of noisy student training.

4.3.6 Results

Table 4.5 presents the experimental results of short-chunk ResNets and Music Tagging Transformers. For both of the models, it summarizes the results of supervised models (the baseline among training methods), models with data augmentation (DA), models with DA and knowledge expan-

Models	#param	ROC-AUC	PR-AUC
ResNet [104]	13.5m	0.9098	0.3525
ResNet+DA	13.5m	0.9141	0.3705
ResNet+DA+KE	13.5m	0.9165	0.3728
ResNet+DA+KD	3.4m	**0.9171**	**0.3742**
Transformer	4.6m	0.9188	0.3775
Transformer+DA	4.6m	0.9191	0.3845
Transformer+DA+KE	4.6m	0.9204	0.3839
Transformer+DA+KD	0.5m	**0.9217**	**0.3889**

Table 4.5: Performance comparison using the CALS MSD split for music tagging.

sion (KE), and models with DA and knowledge distillation (KD).

For both short-chunk ResNet and the Music Tagging Transformer, we observe constant improvements when data augmentation and noisy student training (knowledge expansion) are applied accumulatively. This shows that for both of the architectures, the size of the dataset is a factor that limits the performance of the models.

In Section 4.3.4, we mentioned that our ratio of the semi-supervised set is relatively small. There are two observations that may be related to it. First, unlike a previous work in computer vision [76], we could not observe any performance gain by iterating the noisy student training (i.e., repeating to use a student model as the next teacher model). Second, interestingly, the student model with smaller parameters (models with DA and KD) showed better performance than larger models (models with DA and KE). This would be explained more clearly if the models are trained with a significantly richer dataset, one that is bigger and/or has more diverse data. Unfortunately, we could not run such an experiment due to the lack of a suitable dataset.

4.3.7 Conclusion

In this section, we improved music tagging performance with a semi-supervised scheme: noisy student training. Experimental results indicate that the tagging models can be further enhanced using noisy student training – with either knowledge expansion and knowledge distillation. The training scheme is model agnostic, hence the idea can generalize to any music representation models. Although iterative noisy student training did not show performance gain, this needs to be further tested with larger unlabeled dataset.

4.4 Summary

This chapter explored training schemes beyond supervised learning to enhance model performance. It is already known that transfer learning of music representation from more extensive training data is beneficial [64]. But we also showed that transfer learning is practical when tackling classification tasks with noisy labels. Our approach uses more reliable information (i.e., artist labels) and hand-crafted audio features to generate artist group factors (AGFs). The pretext task of predicting AGFs made the model more robust to report better performance when it is fine-tuned with the target task.

Then we explored noisy student training, a successful semi-supervised learning approach, to incorporate both labeled and unlabeled data during the training process. Experimental results showed model-agnostic performance gain by utilizing larger-scale data. Especially, knowledge distillation using noisy student training reported advanced performance with a smaller number of parameters. It shows the importance of training schemes in music representation learning. Also, it emphasizes the impact of scalable research.

Chapter 5

MULTIMODAL REPRESENTATION LEARNING

5.1 Introduction

Through Chapter 3 and Chapter 4, we investigated advanced music audio representation learning approaches. In this chapter, based on the learned knowledge, we bridge the music audio representation with another modality to form a multimodal embedding space. Especially, we aim at learning multimodal music representation by bridging music audio with language semantics. Multimodal representation learning provides different view points of the same content. Each modality may complement or supplement other modality, as a result, the multimodal embedding can represent the content more informatively [163]. Inclusion of natural language processing in music representation learning may enable enlarged vocabulary for music retrieval. By leveraging tag-level similarity, conventional tag-based music retrieval models can step further beyond fixed vocabulary. Furthermore, we can adopt recent language models [30] to facilitate sentence- or paragraph-level music retrieval, which gets closer towards our ultimate goal: music retrieval with natural language interaction.

This chapter includes the following works:

- Minz Won, Sergio Oramas, Oriol Nieto, Fabien Gouyon, and Xavier Serra, Multimodal Metric Learning for Tag-based Music Retrieval, The International Conference on Acoustics, Speech, Signal Processing (ICASSP) 2021.

- Minz Won, Justin Salamon, Nicholas J Bryan, Gautham J Mysore, and Xavier Serra, Emotion Embedding Spaces for Matching Music to Stories, International Society for Music Information Retrieval (ISMIR) 2021 (**Best student paper**).

5.2 Tag-to-music retrieval

5.2.1 Introduction

Text-based search is one of the most common ways of browsing the internet. This information behavior is also prevalent when exploring music libraries: from querying editorial metadata (e.g., title, artist, album) to high-level music semantics (e.g., genre, mood). To scale the music annotation process, audio-based automatic music tagging has been actively explored by music information retrieval (MIR) researchers [104]. However, this categorical classification has an intrinsic limitation: it can only use a fixed vocabulary. When an out-of-category tag is queried, music tagging models tend to not properly generalize since new tags are not considered during training. In a real world scenario, users query a virtually unlimited amount of music tags. Hence, the music retrieval system needs to be more flexible beyond categorical models.

As opposed to categorical classification models, metric learning aims to construct distance metrics for establishing similarity of data [164, 165]. It can form a similarity metric between two instances from the same modality using shared weights (e.g., Siamese networks [166]) and this can be also easily expanded towards multiple modalities [167, 168]. By jointly learning a multimodal embedding space, metric learning has already demonstrated its suitability for cross-modal retrieval such as image-

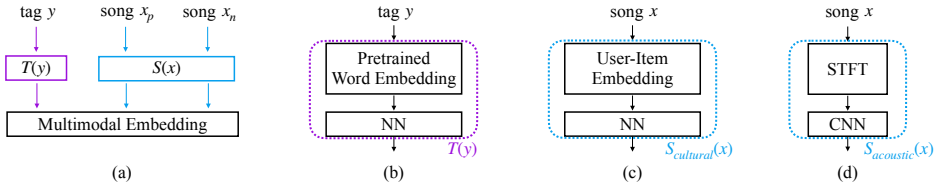


Figure 5.1: (a) Overall architecture of the tag-based music retrieval model. (b) Tag embedding branch. (c) Song embedding branch with cultural information. (d) Song embedding branch with acoustic information.

to-text [169, 167] and video-to-audio [170]. Metric learning facilitates the nearest neighbor search in the embedding space directly, while classification models require a two-step retrieval (i.e., tagging and ranking). Also, metric learning enables abundant vocabulary when pretrained word embeddings are used to represent tags as side information [169, 90].

Recent work in MIR showed the advantage of using metric learning with pretrained word embeddings for audio-based music tagging and classification [90]. Based on the proposed model, we investigate several ideas to successfully introduce metric learning for tag-based music retrieval.

Our contribution in this section is four-fold: (*i*) we show the importance of elaborate triplet sampling, (*ii*) we explore *cultural* and *acoustic* information to represent music, (*iii*) we examine domain-specific word embeddings, and (*iv*) we present a manually cleaned dataset for reproducibility.

5.2.2 Model

Related work

A triplet network [171] is a type of metric learning that uses a triplet loss to fit a metric embedding, where a positive example x_p belongs to the same class as an anchor x_a, and a negative example x_n is a member of a different one. The triplet network is optimized to satisfy $Sim(x_a, x_p) > Sim(x_a, x_n)$, where $Sim(.)$ is a learned similarity metric. As it learns by comparisons, instead of using direct labels, the triplet approach is ex-

pandable to leverage various data sources that are not explicitly labeled. Thanks to its flexibility, deep metric learning with the triplet loss has been actively used to solve a set of diverse MIR problems [168, 90].

Choi et al. [90] proposed a triplet network that learns a multimodal embedding of audio and word semantics. To handle unseen labels, the authors used pretrained GloVe embeddings [42] as side information. An audio embedding branch learns the mapping of the audio input to the multimodal embedding space. And another branch maps pretrained word embeddings to the shared multimodal embedding space. This metric learning model with side information demonstrated its versatility in multi-label zero-shot annotation and retrieval tasks. Since it can perform cross-modal retrieval (i.e., text-to-music), we adopt this architecture design as the backbone of our tag-based music retrieval model.

Architecture overview

Similar to previous work [90], our model is based on two branches. One branch $T(y)$ learns the mapping of tag semantics y to the embedding space, and another branch $S(x)$ learns the mapping of song information x to the shared embedding space — see Figure 5.1-(a). The model is trained to minimize the following loss function L:

$$L = [D(E_a, E_p) - D(E_a, E_n) + \delta]_+, \tag{5.1}$$

where D is a cosine distance function, δ is a margin, and E_a, E_p, E_n are mapped embeddings of anchor tag, positive song, and negative song, respectively. $[\cdot]_+$ is a rectified linear unit. The margin δ prevents the network from mapping all the embeddings to be the same (i.e., $L = 0$ for any inputs). With learnable transformations $T(y)$ and $S(x)$, the equation can be rewritten as:

$$L = [D(T(y_a), S(x_p)) - D(T(y_a), S(x_n)) + \delta]_+, \tag{5.2}$$

where y_a is the anchor tag input, and x_p and x_n are positive and negative song inputs, respectively. The following subsections depict the details of each branch $T(y)$ and $S(x)$.

Tag embedding

Figure 5.1-(b) shows the tag embedding branch $T(y)$. A given tag y passes through the pretrained word embedding model which results in a 300-dimensional vector. By using the pretrained word embeddings, the system can handle richer vocabulary than categorical models. For example, one can expect the system to handle plural forms (*guitar* and *guitars*), synonyms (*happy* and *cheerful*), acronyms (*edm* and *electronic dance music*), and dialectal forms (*brazil* and *brasil*). As our baseline, we use Word2Vec [41] embeddings pretrained with Google News dataset. The tag embedding is input to a neural network which is fully connected to a 512-dimensional hidden layer followed by a 256-dimensional output layer.

Song embedding

Pachet et al. [61] outlined three main types of music information: *editorial*, *cultural*, and *acoustic* — see Section 2.2.6. Most of the previous works in music tagging [104] and multimodal metric learning [168, 90], focused mainly on acoustic information to represent music. In our work, we attempt to operate on not only acoustic information but also cultural information in music retrieval. Cultural information is produced by the environment or culture. One of the most common methods to obtain cultural information is collaborative filtering [172].

The song embedding branch with cultural information $S_{cultural}(x)$ consists of a user-item embedding and a neural network — Figure 5.1-(c). The user-item embedding is obtained by factorizing a user-song interaction matrix. Weighted matrix factorization with the alternating least squares [173] is used, yielding both user and song embeddings of 200 dimensions each. User embeddings are discarded and song embeddings are used as our input. The input of the neural network is fully connected to a 512-dimensional hidden layer followed by a 256-dimensional output layer.

The song embedding branch with acoustic information $S_{acoustic}(x)$ learns audio-based music representation using a convolutional neural net-

work (CNN) — Figure 5.1-(d). According to previous research [104], a simple 2D CNN with 3×3 filters could achieve competitive results to state-of-the-art in music tagging when it uses a short chunk of audio inputs (≈4s). For simplicity, we adopt the short-chunk CNN [104] to train our acoustic embedding.

The model is optimized using Adam [109] with 10^{-4} learning rate, and 10^{-4} weight decay. The model is trained for 200 epochs where 1 epoch includes 10,000 triplets. For input preprocessing, audio files are downsampled to 22.5kHz then converted to mel spectrograms using 1024-point FFT with 50% overlap and 128 mel bands.

5.2.3 Dataset

The Million Song Dataset (MSD) [43] is a collection of metadata and precomputed audio features for 1 million songs. Along with this dataset, the Echo Nest Taste Profile Subset [174] provides play counts of 1 million users on more than 380,000 songs from the MSD, and the Last.fm Subset provides tag annotations to more than 500,000 songs from the MSD. We take advantage of these two subsets of the MSD to build our own dataset. Tags in the Last.fm Subset are very noisy, including 522,366 distinct tags. We performed a cleanup process of the dataset (e.g., merge synonyms or acronyms, fix misspelling) in order to have fewer tags while supported with a reasonable number of annotations. The cleanup process consists of the following steps:

- Filter out all tracks not included in the MSD Taste Profile.
- Filter out all tag annotations with a Last.fm tag score of 0 (Last.fm tags in the original dataset come with a score between 0 and 100).
- Filter out all tracks with more than 20 tags (we assume that annotations in tracks with too many tags are less reliable).
- Preprocessing of tags: remove punctuation, normalize expressions (e.g., and, &, 'n'), and remove irrelevant suffixes (e.g., music, song, tag).

- Group all tags with the same preprocessed lemma and name the group with the most common of the involved tags.

- Select all tag groups with annotations for at least 100 tracks.

The final dataset contains 500 tag groups (from now on we call these groups "tags"), which yields 1,352 distinct Last.fm tags. These 500 tags are then manually classified in a lightweight taxonomy of 7 classes (genre, mood, location, language, instrument, activity, and decade). 158,323 distinct tracks are tagged with these 500 tags with an average of 3.1 tags per track and each track has user play counts. Table 5.1 shows the category-wise distribution. We release the final dataset as the *MSD500*.

Class	**Number of tags**
genre	294
mood/character	94
location	36
language/origin	34
instrument	21
activity	14
decade	7

Table 5.1: *MSD500* number of tags per class

In this chapter, we use two different subsets of the proposed dataset which are *MSD100* and *MSD50*. Music tags are highly skewed towards few popular tags and handling this skewness is another big topic in data-driven approaches. Models are optimized to predict more popular tags in the training set while evaluation metrics are averaged over tags. To avoid the undesired effect of the high skewness, we only use the top 100 tags in our experiments which results in 115k songs (*MSD100*).

Although we have user information in our dataset, the interaction counts are not scalable compared to industry standards [175]. This may underrepresent the predictive power of cultural information. Hence, we build another subset which includes 39,402 songs with Last.fm tags and

Metrics	Random	Balanced	Balanced-weighted
MAP	0.1658	0.1675	**0.1852**
P@10	0.2990	0.3160	**0.3500**

Table 5.2: Performance of different samplings (MSD100).

user-item embeddings from more than 100B in-house user explicit feedback. In this case we only use the top 50 tags (*MSD50*) because the size of the dataset became smaller during the mapping process. As the in-house user feedback includes sensitive information, we only release the song IDs and their tags of the *MSD50*. All data splits have been done at an artist level to avoid unintentional information leakage.

5.2.4 Experiments

In this section we introduce three experiments which can be critical to enhance our metric learning approach for tag-based music retrieval. All models are evaluated with mean average precision (MAP) over the labels and precision at 10 (P@10). Reproducible code and dataset are available online. [1]

Sampling matters

The number of possible triplets grows cubically as the number of observations grows. Thus, triplet sampling is crucial in deep metric learning [176], as it matters equally or more than the choice of loss functions. In this subsection, we explore three different sampling methods: random sampling, balanced sampling, and balanced-weighted sampling.

Random sampling randomly chooses one song to generate an anchor-positive pair. Then a negative example is randomly sampled from a set of songs without the anchor tag. With this method, more popular tags are more likely to be sampled as the anchor tag. Also, songs with less popular

[1] https://github.com/minzwon/tag-based-music-retrieval

tags are less likely to be sampled as negative examples due to their small numbers.

To alleviate this problem, the balanced sampling method uniformly samples an anchor tag first and then select a positive song. Minor tags may have equal possibilities to popular tags to be sampled as an anchor tag. By sampling negative examples from the batch of the positive songs, we can also expect more balanced tag distribution of negative examples.

For more efficient training, various triplet sampling methods have been proposed such as hard negative mining [177], semi-hard negative mining [178], and distance weighted sampling [176]. We combine the distance weighted sampling [176] with the aforementioned tag balancing method (i.e., balanced-weighted sampling). As in balanced sampling, we select an anchor tag and a positive song. From the batch of positive songs, we sample negative examples. Sampling weights are inversely proportional to their cosine distances from the anchor tags in the embedding space. Thus, more informative (harder) negative examples are more likely to be sampled while not loosing semi-hard and soft negative examples.

As shown in Table 5.2, balanced-weighted sampling outperforms other sampling methods. This proves that sampling matters for training our tag-based music retrieval model. Note that here we only used acoustic information for the song embedding to control the experiment. From now on, the following experiments use the balanced-weighted sampling method.

Acoustic and cultural music representation

We believe certain groups of tags are more related to acoustic information while others may be more culturally relevant. A tag *piano*, for example, can be predicted using the user-item matrix if there is a specific group of users who heavily listened to songs with piano. However, originally, the tag *piano* is associated with acoustic information. When there is a song beloved by the aforementioned user group, if we only use cultural information, the song can be regarded as piano music even when no piano can be acoustically perceived in the song. As another example, a tag *K-pop* can be predicted based on acoustic information since there are common

Metrics	MSD100			MSD50		
	Cul-E	Acoustic	Concat	Cul-E	Cul-I	Acoustic
MAP	0.1155	**0.1852**	0.1775	0.2163	**0.4719**	0.3062
P@10	0.3200	**0.3500**	0.3120	0.4500	**0.6380**	0.4680

Table 5.3: Performance of cultural and acoustic models.

acoustic characteristics of *K-pop*. However, if the song is not from Korea and is not being consumed in Korea, it should not be tagged as *K-pop*. To investigate the capability of two different information sources, we train our metric learning model with cultural information only and acoustic information only: $S_{cultural}$ and $S_{acoustic}$, respectively.

As shown in Table 5.3, the acoustic model outperforms the cultural model on *MSD100*. However, if we take a closer look at category-wise scores, the cultural model shows its strength in *language/origin* and *location* tags (Figure 5.2). This supports our hypothesis that the modality selection has to be associated with its original source of information. But a more important factor than the information source is the size and quality of available data. In Table 5.3 (*MSD50*), we have two different cultural models Cul-E and Cul-I, which use the EchoNest Taste Profile and our in-house user explicit feedback, respectively. Since our in-house data are of industry scale and explicit, they are richer than the publicly available data. As cultural information becomes richer (Cul-I), the cultural model outperforms the acoustic model. In addition, we observed that the cultural model with richer information (Cul-I) is superior in every tag category including *genre* and *mood*. As observed, acoustic and cultural models show different strengths, but the foremost important factor of the modality selection is the size and quality of available user-item interactions and audio data. We also experimented with a hybrid model with simple concatenation of cultural and acoustic embeddings but it did not improve results (Table 5.3-Concat).

Tag	GoogleNews	Domain-specific
Jungle	jungles, dense_jungle, dense_jungles, rainforest, thick_jungles, Amazon_jungle, Amazonian_jungle, steamy_jungles, hilly_jungle, swamps	**breakbeat, dub, drum_n_bass, drum'n'bass, grime, deep_house, ragga, dubstep, acid, acid_house**
House	houses, bungalow, apartment, bedroom, townhouse, residence, mansion, farmhouse, duplex, appartment	**deep_house**, kitchen, **club**, **jungle**, **rave**, **warehouse**, **parties**, **tech-house**, **lounge**, **ibiza**
Country	nation, continent, region, thecountry, world, coun_try, United_States, countrys, coutnry, counry	**traditional_country**, **bluegrass**, **americana**, **nashville**, **folk**, **western_swing**, **rockabilly**, **cajun**, **gospel**, **hillbilly**
Metal	**Metal**, metals, aluminum, steel, stainless_steel, precious,_metal, copper, metallic, jacketed_bullet, titanium	**heavy_metal, death_metal, thrash, thrash_metal, power_metal, extreme_metal, metalcore, speed_metal, progressive_metal, black_metal**
Chill	chilly, cold, chilled, chills, shivers, shiver, warm, frigid, frosty, balmy	**chill_out**, relax, chilled, kick_back, relaxing, **chill-out**, **chilled_out**, **downtempo**, **down_tempo**, unwind
Brazilian	Uruguayan, Brazil, Argentine, Argentinean, Brazilan, Brazillian, Portuguese, Sao_Paulo, Peruvian, Brazilians	cuban, **mpb**, latin_american, portuguese, colombian, **bossa_nova**, brasilian, latin, argentinian, **samba**
Smooth jazz	N/A	**contemporary_jazz, jazz, latin_jazz, acid_jazz, new_age, neo-soul, easy_listening, soul-jazz, kenny_g, bossa_nova**
Deep house	N/A	**progressive_house, breakbeat, tech-house, downtempo, tech_house, minimal_techno. electro_house, drumńbass, drum_n_bass, uk_garage**

Table 5.4: Nearest words in GoogleNews and domain-specific word embeddings. Music-related words are emboldened.

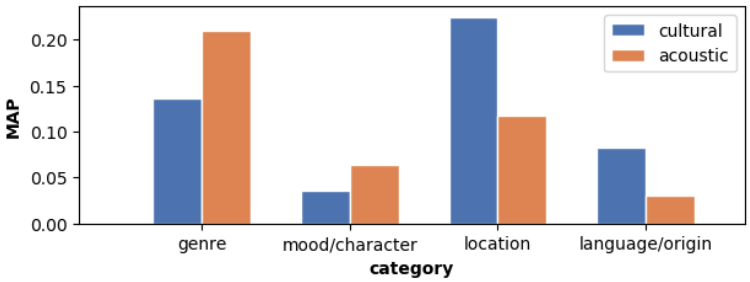


Figure 5.2: Category-wise MAP on *MSD100*.

Domain-specific word embeddings

We use pretrained Word2Vec [41] embeddings as a part of our tag branch $T(y)$. Since they are trained with Google News, the embeddings are not expected to have musical context.

We pretrain our own word embeddings with musical text data. We use the corpus of text from the subtask 2B of the SemEval-2018 Hypernym Discovery Task [2]. It contains an already tokenized 100M-word corpus including Amazon reviews, music biographies, and Wikipedia pages about theory and music genres. We train a Word2Vec model on this corpus with a window of 10 words yielding word embeddings for unigrams, frequent bigrams and trigrams of 300 dimensions.

We could not discover any quantitative performance gain by using our domain-specific word embeddings. However, as shown in Table 5.4, the domain-specific word embeddings may include more musical context. For example, for the unseen query *jungle*, a model with domain-specific embeddings could successfully retrieve relevant items while conventional embeddings could not. Also, domain-specific music corpora include frequent bigrams and trigrams, such as *deep house* or *smooth jazz*, which are not typically captured in word embeddings trained on general text corpora. More qualitative examples are included in our online repository.

[2]https://competitions.codalab.org/competitions/17119#learn_the_details-terms_and_conditions

5.2.5 Conclusion

In this section, we explored three different ideas to enhance the quality of metric learning for tag-based music retrieval. Balanced-weighted sampling could successfully improve the evaluation metrics. Cultural and acoustic models showed different strengths based on the information source of the given tag but the foremost important factor is the size and quality of available data. Finally, domain-specific word embeddings showed their suitability for music retrieval by including more musical context.

As future work, in-depth comparison of acoustic and cultural models is necessary to better understand how the size and the quality of data affect the results. Also, a hybrid method of fusing acoustic and cultural information should be explored. Finally, to meet real-world expectations, multi-tag retrieval systems have to be considered.

5.3 Sentence-to-music retrieval

5.3.1 Introduction

Content creators, both amateur and professional alike, often use music to enhance their storytelling due to its powerful ability to elicit emotion [3]. For example, when dissonant music is added to a horror movie, it can amplify the scary mood of the story line. Similarly, cheerful music can emphasize the excited mood in a scene of a birthday party. Matching text and music to create a narrative, typically requires tediously browsing large-scale music collections, significant experience, and musical expertise. In this section, we therefore address the problem of automatically matching music to text as shown in Figure 5.3.

We formalize this task as a cross-modal retrieval problem [179] and focus on matching long-form text (multiple sentences, paragraphs) to music. For queried sentences like books and scripts, we seek to retrieve matching music for applications such as podcasts, audio books, movies, and film. To facilitate cross-modal retrieval, a common approach is to

[3] We use the terms *emotion* and *mood* interchangeably following previous work [8].

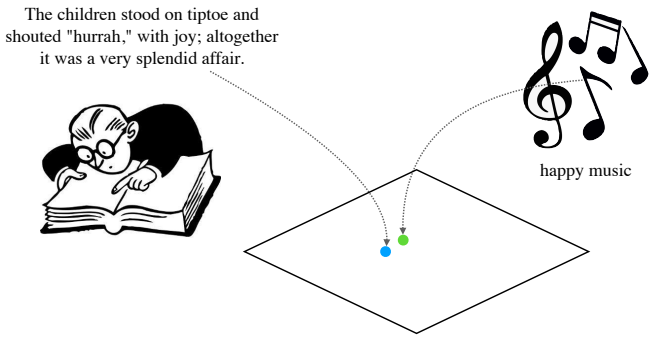


Figure 5.3: Cross-modal text-to-music retrieval using an aligned, multimodal embedding space.

first perform feature extraction to convert each data modality into an embedding space. Then, the different embedding spaces must be matched to bridge the *modality gap* by somehow aligning their different distributions [179]. Once aligned, (fast) nearest neighbor search can be used for retrieval.

Various methods have been proposed for cross-modal feature extraction and alignment. For example, canonical correlation analysis has been used to bridge the modality gap [180] as well as modern deep learning techniques that learn common representation spaces [181, 182]. Such methods can be categorized into four groups: unsupervised, pairwise-based, rank-based, and supervised methods [183]. Among these, supervised methods are the most straightforward and in theory can take advantage of existing labeled datasets (e.g., labels of *happy*, *sad*) and themes (e.g., *party*, *wedding* with corresponding text and music). Difficulties, however, immediately arise because of mismatched dataset taxonomies (vocabularies) per modality, making it challenging to use standard techniques directly.

Therefore, in this work we focus on the task of emotion-based text- (e.g. sentences, paragraphs) to-music retrieval, and investigate how we can best perform cross-modal retrieval with heterogeneous dataset taxonomies. To the best of our knowledge, this problem has not been previ-

ously addressed and could be beneficial to media content creation applications. We propose six different deep learning strategies to extract relevant features and bridge the modality gap between text and music including (*i*) classification (*ii*) multi-head classification (*iii*) valence-arousal regression (*iv*) Word2Vec regression (*v*) two-branch metric learning and (*vi*) three-branch metric learning. We then evaluate each approach on multiple text and music datasets, report objective results via precision at five and mean reciprocal rank, and conclude with qualitative analysis and discussion. Our results show that our valence-arousal-based method is a powerful baseline for emotion-based cross-modal retrieval, but that our three-branch metric-learning approach is comparable, more general, and does not require manually engineered valence and arousal mappings.

5.3.2 Related Work

Text Emotion Classification

Text emotion classification methods or the task of predicting emotion from text can be divided into three categories: lexicon-based models, traditional machine learning models, and deep learning models. Lexicon-based models take advantage of pre-defined emotion lexicons, such as NRC EmoLex [184] and WordNet-Affect [185] to match keywords. Traditional machine learning approaches recognize emotions using algorithms such as support vector machine (SVM) [186] and Naive Bayes [187]. Finally, deep learning models use deep sequence models such as gated recurrent unit (GRU) [188], bidirectional long-short term memory (BiLSTM) [189], and Transformers [190]. Most recently, Transformer models [30, 191, 192] have become quite prevalent. Such models take advantage of transfer learning, are commonly pre-trained to learn language representation with large datasets, and then applied to various downstream tasks including question and answer systems as well as emotion recognition [190].

Music Emotion Classification

Music emotion classification or the task of predicting emotion from music audio is commonly divided into conventional feature extraction and prediction approaches [193, 194, 195], and end-to-end deep learning approaches [196, 103]. Deep learning approaches have become most prevalent and commonly frame emotion recognition as a multi-class or multi-label auto-tagging classification problem [115, 57, 112, 108, 197]. Recently, multiple music tagging models were evaluated in a homogeneous evaluation pipeline [104] and found three design recommendations for automatic music tagging models: (1) use mel-spectrogram inputs, (2) use 3×3 convolutional filters, and (3) use short-chunk audio inputs with small hop sizes and max-pooling. Based on this, a model using mel-spectrogram inputs and convolutional neural networks with focal loss [198] won the MediaEval 2020 Emotion-and-Theme-Recognition-in-Music-Task[4] [199].

Valence-arousal Regression & Word Embeddings

Beyond classification, previous works [200, 201] suggest that regression approaches can outperform classification approaches in music emotion recognition. Here, researchers use the well-known valence-arousal emotion space [202, 203] where valence represents positive-to-negative emotions, and arousal indicates the intensity of the emotions. These annotations can be collected by human annotators directly [200] or by mapping existing mood labels into the valence-arousal space using pre-defined lexicons [103, 204].

As an alternative to using the manually annotated valence-arousal space, we can obtain tag (mood) embeddings in a more data-driven fashion. Pre-trained word embeddings, such as Word2Vec [41] and GloVe [42], represent words as vectors by learning word associations from a large corpus. These embedding spaces use the cosine similarity as a measure of semantic similarity. Recent works [90, 91] show the suitability of pre-

[4]https://multimediaeval.github.io/2020-Emotion-and-Theme-Recognition-in-Music-Task

trained word embedding in music retrieval and that the embedding can include more music related context by training it with music related documents [91, 205].

Cross-modal Retrieval

Instead of targeting a pre-defined embedding space, multimodal metric learning models aim at learning a shared embedding space in which semantically similar items are close together while dissimilar items are far apart in the embedding space. Unsupervised approaches leverage co-occurrence information. For example, when we collect user-created video from the web, the video and audio streams are synchronized, and this correspondence can be exploited for representation learning [206, 207]. On the other hand, supervised methods learn discriminative representations by exploiting annotated labels. Here, data from different modalities are used to train models such that data points with the same label should be close while data with different labels should be far apart. Metric learning is also used for bridging the modality gap between text and audio, such as keyword spotting [208], text-based audio retrieval [209, 210], and tag-based music retrieval [90, 91] in both supervised and unsupervised ways.

Two branch metric learning [211] is one of the most prevalent architectures for cross-modal retrieval. It consists of two branches where each branch extracts features from each modality and maps them into a shared embedding space. When optimized with a conventional triplet loss (e.g. anchor text, positive song, negative song), however, the model loses neighborhood structure within modalities. To alleviate this issue, previous work [167] added structure-preserving constraints by using additional triplet losses within modalities (e.g., anchor text, positive text, negative text).

5.3.3 Models

Cross-modal retrieval comprises two parts: feature extraction and bridging the modality gap. Our text and music embeddings, E_{text} and E_{music}

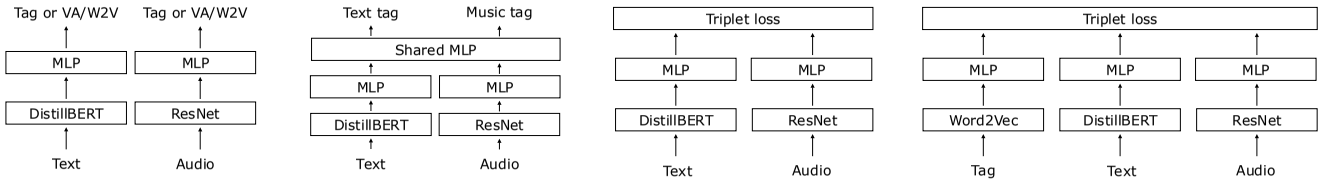


Figure 5.4: Model architectures. (a) Classification and regression models (b) Multi-head classification model with shared weights (c) Two-branch metric learning (c) Three-branch metric learning.

respectively, are defined as follows:

$$\begin{aligned} E_{text} &= M(P_{text}(x)) \\ E_{music} &= M(P_{music}(x)) \end{aligned} \tag{5.3}$$

where P is a pre-trained model to extract features from each modality and M is a multilayer perceptron (MLP) to map them to a multimodal embedding space.

Pre-trained Models for Feature Extraction

In our work, we leverage the DistilBERT [192] transformer model for text analysis, which is a compact variant of the popular BERT transformer model [30, 192]. We use a pre-trained model from the Huggingface library [212].

For the music representation model P_{music}, we use a CNN with residual connections that are trained with mel-spectrograms (ResNet) [104]. Due to its simplicity and high performance, it is a broadly used architecture not only in music but also in general audio representation learning. Our ResNet consists of 7 convolutional layers with 3×3 filters followed by 2×2 max-pooling. The model is pretrained with the MagnaTagATune dataset [54]. Both pre-trained models are updated during the training process so that they can adapt to the data.

Classification

As a starting point, we train two separate mood classification models for text and music (Figure 5.4-(a)). Then the model returns mood predictions and their likelihood with softmax. From the predicted text mood, songs are re-ranked based on their likelihoods of the text mood. However, this classification approach has an inherent limitation- the model cannot bridge the modalities when they have different mood taxonomies.

Multi-head Classification with Shared Weights

Multi-head model is similar to the classification model but it shares a 3-layered MLP for multimodal fusion in it (Figure 5.4-(b)). Since the model shares the weights across different modalities, it can predict the mood in different taxonomies by switching the classification head. We included this model to see if the shared MLP can generalize across modalities.

Regression

Following previous work [103], we reformulate the classification task as a regression problem. By using NRC VAD Lexicon [204], emotion labels can be mapped to the valence-arousal space. However, this mapping process is hand-crafted and also they cannot handle bi-grams or tri-grams since the lexicon was created in a word-level. In addition to leveraging the valence-arousal space, we also experiment with a Word2Vec [41] embedding which was pre-trained with music related text [91]. This data-driven space supports a larger vocabulary, including bi-grams and tri-grams, and is thus more flexible.

Regression models are trained separately for each modality (Figure 5.4-(a)). Then the nearest items are retrieved based on their distance in the embedding space. Note that, distance metrics are Euclidean distance for the valence-arousal space, and cosine distance for the Word2Vec space. However, regression is a one-way optimization, i.e., optimizing text or mood into the pre-defined word embedding space. In this case, neighborhood structure within each modality can be ignored. For example, music with *angry* and *exciting* can share similar acoustic characteristics. However, if two words are far apart in Word2Vec space, this similarity cannot be considered by regression. This obstacle motivates us to learn a shared embedding space in a data-driven fashion using metric learning.

Metric Learning

Finally, we explore metric learning, which is a fully data-driven approach that solves the cross modal text-to-music retrieval in an end-to-end man-

ner. Metric learning is optimized to minimize a triplet loss $\mathcal{T}$:

$$\mathcal{T}(E_a, E_p, E_n) = [D(E_a, E_p) - D(E_a, E_n) + \delta]_+ \tag{5.4}$$

where D is a cosine distance function, δ is a margin, and E_a, E_p, E_n are embedding of anchor, positive, and negative examples, respectively. $[\cdot]_+$ is rectified linear unit. Following conventional metric learning models for cross-modal retrieval, we implement a two-branch metric learning model [211] (Figure 5.4-(c)) that optimizes the loss function L,

$$L = \mathcal{T}(E^a_{text}, E^p_{music}, E^n_{music}). \tag{5.5}$$

However, with the triplet function, neighborhood structure or data distribution within modalities can be lost. Structure-preserving constraints [167] can alleviate the issue but our problem is different from the case, since we have different taxonomies across the modality which includes many non-overlapped moods.

To take advantage of different mood distribution of different modalities, we investigate metric learning model with three branches (Figure 5.4-(d)) that results in three triplet loss functions. Each loss function is designed to optimize tag-to-text, tag-to-music, and text-to-music triplet losses as following:

$$\begin{aligned} L_{text} &= \mathcal{T}(E^a_{tag}, E^p_{text}, E^n_{text}), \\ L_{music} &= \mathcal{T}(E^a_{tag}, E^p_{music}, E^n_{music}), \\ L_{cross} &= \mathcal{T}(E^a_{text}, E^p_{music}, E^n_{music}). \end{aligned} \tag{5.6}$$

The model learns a shared mood space between Word2Vec embedding and text embedding with a loss L_{text}, and a shared mood space between Word2Vec embedding and music embedding with a loss L_{music}. Finally, they are bridged together with a cross-modal triplet loss L_{cross}. We refer to this model as three-branch metric learning.

Since text and music have different vocabularies in our scenario, for both two-branch and three-branch metric learning, we regard the nearest tags in pre-trained Word2Vec space as positive pairs in cross-modal triplet sampling (Table 5.5). We used distance-weighted sampling [176] for more efficient negative mining following our previous work in Section 5.2.

5.3.4 Experimental Design

Text Datasets

Alm's affect dataset [213] includes 1,383 sentences collected from books written by three different authors: B. Potter, H.C. Andersen, and the Brothers Grimm. 1,207 sentences in the dataset are annotated with one representative emotion among five: *angry*, *fearful*, *happy*, *sad*, and *surprised*. To avoid unintended information leakage, we decided to split data in an author-level. 1,040 sentences by the Brothers Grimm and H.C. Andersen were used for training and 167 sentences by B. Potter were used for validation and test.

ISEAR dataset [214] is a corpus with 7,666 sentences that are categorized into one of seven emotion: *anger*, *disgust*, *fear*, *joy*, *sadness*, *shame*, and *guilt*. Each sentence describes certain antecedents and those are associated with according reactions (emotions). We split the dataset in a stratified manner with ratio of 70% train, 15% validation, and 15% test set.

Music Dataset

There are multiple datasets for music emotion recognition such as the Million Song Dataset (MSD) subset [215, 216], the MTG-Jamendo mood subset [40], and the AudioSet mood subset [135]. Before we choose our dataset, we run classification experiments for each subset. AudioSet subset returned the highest accuracy, which means the labeled emotions are predictable with our ResNet model. One possible reason for this result is that unlike other datasets, emotion labels of AudioSet subset are exclusive, having a single emotion label per song. This is also beneficial since we can map each song directly to the valence-arousal space or word embedding space using emotion lexicons or Word2Vec model, respectively. Otherwise, to handle multiple tags, we need to average their embedding vectors as previous researchers did [103]. For these simplicity and reliability reasons, we use AudioSet mood subset.

AudioSet [135] mood subset consists of 16,995 music clips collected

Original	VA	W2V	Manual
anger	angry	angry	angry
fearful	sad	scary	scary
happy	happy	happy	exciting, funny, happy
sad	sad	sad	sad
surprised	exciting	happy	exciting
anger	angry	angry	angry
disgust	angry	angry	angry, scary
fear	angry	angry	scary
guilt	sad	angry	angry, sad
joy	exciting	tender	exciting, funny, happy
sadness	sad	tender	sad
shame	angry	sad	angry, sad

Table 5.5: Similar moods from Alm's dataset (upper) and ISEAR dataset (lower). Original is from text mood taxonomy and mapped tags are from music dataset.

from YouTube and each audio clip is 10-second long. The dataset is categorized into 7 mood categories: *happy*, *funny*, *sad*, *tender*, *exciting*, *angry*, and *scary*. The dataset is provided with a training set of 16,104 clips and an evaluation set of 540 clips.

Evaluation

We use two evaluation metrics: Precision at 5 (P@5) and Mean Reciprocal Rank (MRR). However, since our text and audio datasets use different taxonomies, we need a mapping between the different vocabularies in order to compute the metrics directly. Thus, we map the text emotion taxonomy to the music emotion taxonomy — see Table 5.5. We introduce three possible mappings: (1) mapping based on the Euclidean distance between emotion labels in the valence-arousal space (VA), (2) the cosine distance between emotion labels in Word2Vec space (W2V), or (3) direct manual mapping of emotion labels. Given these mappings, we compute P@5 and MRR. Another challenge is the label distribution in our datasets,

Methods	Alm's dataset						ISEAR dataset					
	VA		W2V		Manual		VA		W2V		Manual	
	P@5	MRR	P@5	MRR	P@5	MRR	P@5	MRR	P@5	MRR	P@5	MRR
Classification	0.2161	0.2436	0.1861	0.2157	0.2161	0.2436	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Multi-head Classification	0.2819	0.4181	0.1271	0.1381	0.3446	0.5304	**0.3440**	0.5084	0.3325	0.3625	0.3551	0.4803
V-A Regression	**0.4325**	**0.6282**	0.4125	0.5749	**0.6100**	**0.7398**	0.3018	**0.5247**	0.1866	0.3709	**0.6218**	0.7075
W2V Regression	0.3960	0.5010	0.4613	0.5591	0.5413	0.6363	0.3008	0.3829	0.4164	0.4908	0.5527	**0.7668**
Metric Learning (2 branch)	0.3399	0.3778	0.4897	0.5239	0.5374	0.5579	0.2695	0.3287	0.3951	0.4336	0.4438	0.6175
Metric Learning (3 branch)	0.3574	0.4348	**0.5095**	**0.5863**	0.5156	0.5880	0.2591	0.3445	**0.4317**	**0.4953**	0.6019	0.6675

Table 5.6: Retrieval scores

which is unbalanced. This can lead to over-optimistic results if the model performs well on the majority class, even if it performs very poorly on less common labels in the test dataset. To alleviate this problem, we compute the macro-P@5 and macro-MRR, i.e., we compute the metrics per class (emotion label) then average the per-class results. Henceforth we will use P@5 and MRR to denote *macro* P@5 and MRR, respectively.

Regression models are optimized to reduce mean squared error and metric learning models are optimized with the triplet losses detailed in Section 5.3.3. We use the Adam optimizer with learning rate 0.0001 for all models. Audio inputs are resampled into 16 kHz and then converted to 128-bin mel-spectrograms via a 512-point FFT with 50% frame overlap. Implementation details are available online [5].

5.3.5 Results

Quantitative Results

The retrieval results for the different proposed models, using our three different proposed vocabulary mappings (VA, W2V, Manual), for our two text datasets, are presented in Table 5.6. First, we see that the classification model fails in cross-modal retrieval. Since there are only two emotions in common between Alm's dataset and AudioSet (i.e., *happy* and *sad*), text inputs with other emotions will not have any retrieval result. Furthermore, there's no common emotion between ISEAR dataset and AudioSet, hence P@5 and MRR are zero in this case. Classification models can be powerful when there are exactly identical or partially overlapped vocabularies, but since it is less likely in real-world data, classification approach is less desirable for cross-modal retrieval.

The multi-head classification model also performs worse than other regression and metric learning models. Some metrics look optimistic but when we check the confusion matrix of the multi-head classification model, it constantly predicts one or two specific emotions (e.g., predict *angry* for any type of input) no matter what the input is. This means the shared MLP

[5] https://github.com/minzwon/text2music-emotion-embedding.git

cannot generalize across different modality heads.

The regression model using valence-arousal consistently shows the best metrics as already proven in previous single-modality emotion recognition works [200, 201]. Since the space is carefully designed and the tag-to-space mapping process has been done manually [204], the valence-arousal regression suits our cross-modal retrieval task. However, this method cannot generalize to other datasets that possibly have some tags that do not have manual tag-to-space mapping. Word2Vec regression is suitable in that case. It shows slightly lower but comparable retrieval performance and it can handle abundant vocabulary, even bi-grams and tri-grams, without a manual mapping process.

Finally, we assess the performance of metric learning. Instead of predicting manually defined or pre-trained embeddings, metric learning aims at learning a shared embedding space across different modalities. Both two-branch and three-branch approaches claim their suitability for cross-modal retrieval, and the three-branch metric learning model consistently outperforms the two-branch model by leveraging the relationship of tag-to-text and tag-to-music within each modality.

Qualitative Results

To further investigate the characteristics of various embedding spaces, we visualize them with 2D projection—Figure 5.5. Due to limited space, we only visualize embedding spaces with Alm's dataset and AudioSet mood subset. Note that they are all predicted embeddings using the test set. Except valence-arousal space (first row), which is already 2D, high dimensional embedding spaces are projected to a 2D space using the uniform manifold approximation and projection (UMAP) [217]. We use UMAP since it preserves more of the global structure compared to tSNE [218]. In the projection process, we first fit the UMAP with one modality (in our figure: music), then projected other embeddings (in our figure: tag and text) into the fitted 2D space.

First of all, for both the Word2Vec embedding space and the metric learning space, relevant moods from different taxonomies are neighbor-

ing together in the embedding space. This is natural for the Word2Vec space because each modality is fitted to optimize the pre-defined word embeddings. But this neighboring also can be found in metric learning space. In Figure 5.5-(g) and (h) for example, *anger* from text and *angry* from music are together, and *fearful* from text and *scary* from music are together. Note that Figure 5.5-(e) and (f) do not have word embeddings since the two-branch metric learning model does not have a branch to map the mood tags into the embedding space.

One of our main motivations to use metric learning with three branches is to preserve neighborhood structure within modalities. Since Word2Vec regression is a one-way optimization, their embeddings are very discriminative (Figure 5.5-(c)). Also, the two-branch neural network does not have any means to learn the neighborhood structure of each modality. Especially, as shown in Table 5.5, when two-branch metric learning uses the mapping of Alm's mood into AudioSet mood with Word2Vec similarity, *exciting* and *tender* from music are not being used in training. If we compare Figure 5.5-(f) and (h), *exciting* music in (h) are more continuously distributed between *angry* and *happy* while they are simply with *happy* in (f). Also, when we compare text embeddings (see (e) and (g)), *surprised* is continuously distributed between *anger* and *happy* in (g) but not in (e). This continuity between music and text can be found in the manually annotated valence-arousal space (see (b) and (a), respectively), which means the proposed three-branch metric learning model preserves neighborhood structure within modalities in the learned multi-modal embedding space. We summarize all the introduced characteristics in Table 5.7.

5.3.6 Conclusion

In this work we tackled the task of matching music to text with the goal of allowing users to enhance their text-based stories with music that matches the mood of the text. We formulated the problem as a cross-modal text-to-music retrieval problem, and identified the lack of a shared vocabulary as a key challenge for bridging the gap between modalities. To address this challenge, we proposed and investigated several emotion em-

Model	Retrieval	Distribution	Mapping
Classification	fail	.	.
Multi-head classification	fail	.	.
V-A regression	success	continuous	manual
W2V regression	success	discriminative	data-driven
Metric learning (2 branch)	success	discriminative	data-driven
Metric learning (3 branch)	success	continuous	data-driven

Table 5.7: Characteristics of different models

bedding spaces, both manually defined (valence/arousal) and data-driven (Word2Vec and metric learning), to bridge between the text and music modalities. Our experiments showed that by leveraging these embedding spaces, we were able to facilitate cross modal retrieval successfully. We showed that the carefully designed valence-arousal space can bridge different modalities, but this can be also achieved via data-driven embedding spaces. Especially, our proposed three-branch metric learning model preserves the neighborhood structure of emotions within modalities. By leveraging data-driven embeddings, our approach has the potential of being generalized to other cross-modal retrieval tasks that require broader or completely different vocabularies.

5.4 Summary

This chapter covered two cross-modal retrieval approaches to bridge music semantics with linguistic semantics. The first approach introduced metric learning models to enable free-form tag-based music retrieval by using a pretrained word embedding space. Impacts of sampling strategies, cultural and acoustic information, and domain-specific word embeddings are introduced. Users can search relevant music with the metric learning model without separate ranking algorithms.

The second approach allows sentence- and paragraph-level inputs to retrieve music. In this approach, we utilized emotion labels to bridge different modalities due to the lack of paired data between text and mu-

sic. More precisely, it introduces sentence-to-music retrieval models when there is no paired data and they have different label taxonomies. Regression of pre-defined emotion embedding space (valence-arousal) claimed a strong baseline, but a more data-driven metric learning approach also reported comparable results while preserving neighborhood structures within the modality. Although this work relies on emotion labels to optimize the model, the proposed metric learning approach showed the potential of bridging sentence- / paragraph-level text semantics with music semantics. If there are appropriate data, one can expect the model to perform music captioning or natural language text-to-music retrieval.

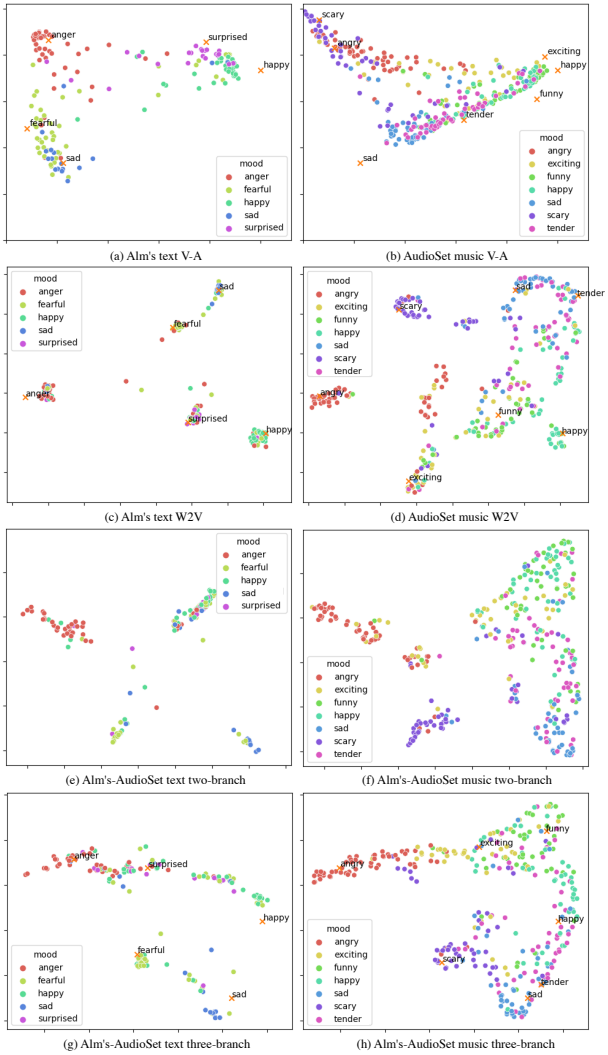


Figure 5.5: Valence-arousal embedding (first row), UMAP of Word2Vec embedding (second row), UMAP of shared embedding space from two-branch metric learning (third row), and UMAP of shared embedding space from three-branch metric learning (fourth row).

Chapter 6

CONCLUSIONS

6.1 Summary of the research

This dissertation explored music representation learning approaches to enhance music classification and retrieval. Chapter 3 revisited various deep model architectures under a unified evaluation pipeline. Assumption-free 2D CNN with 3×3 filters claimed a strong baseline when it is trained with short audio excerpts (short-chunk ResNet [104]). Inclusion of domain knowledge in architecture design brings benefits when the dataset is small, but assumption-free models outperform as the size of the data grows. However, a minimal assumption (i.e., mel spectrogram) is yet beneficial at the current scale. We also proposed an advanced front end design (Harmonic tensors [108]) and a back end architecture (Music Tagging Transformer [113]) to learn more robust representation with better generalizability. In Chapter 4, we further improve the classification performance using transfer learning and noisy student training. As a result, the proposed architecture and training scheme could claim a new state-of-the-art in automatic music tagging. Finally, we bridged the music audio representation with natural language semantics in Chapter 5 to form multimodal embedding spaces. The multimodal spaces enable tag-to-music retrieval beyond fixed vocabulary and automatic matching of suitable music to stories based on moods.

6.2 Limitations

Although the introduced models and training schemes report remarkable performances in music classification, the representation models are yet behind humans' music perception. As we partially or entirely adopt CNNs in our architecture design, the models possibly have an inherent texture bias. As reported in previous research [219], CNNs are biased toward texture. The texture is often interpreted as *timbre* of music. That means timbre transformation or other adversary intent [25] can lead the model to make totally different predictions. Data augmentation can alleviate the issue when known deformations are applied during the training, but we cannot assure the model can generalize to unseen types of deformation. Also, the transformer can alleviate the issue by sequence modeling, but this needs to be demonstrated through careful studies.

Another bias that we need to care about is data bias. Our deep representation models learn from data. If the given dataset is biased, the model will learn the bias. For example, the Ballroom dataset [44] is originally designed to classify different rhythmic patterns. However, those classes in the dataset can be easily distinguished by the tempo. In this case, the model will learn the tempo bias and cannot generalize to the rhythmic patterns in different tempo [220]. Also, the datasets that we used in this research (e.g., MTAT [54], MSD [43], and MTG-Jamendo [40]) are biased toward western music. We cannot assure the models trained with those datasets can generalize to music from other cultural backgrounds. We need to consider de-biasing approaches to build more generalizable representations.

Finally, the proposed multimodal spaces for matching music-to-stories in Chapter 5 yet rely on manual mood labels to train the model. This hinders scalable data-driven research in multimodal representation learning. When a dataset is small, the model is prone to be biased, although we actively use transferred representations for both text and music. Semi-supervised learning can be considered, and other external data can be actively included. We believe the multimodal embedding spaces have a huge potential when they are trained at scale.

6.3 Future research

6.3.1 Self-supervised learning

This dissertation mainly focused on supervised learning and explored some semi-supervised approaches to learn music representation. Although the proposed approaches report significant improvements, the models cannot generalize when it is exposed to inputs outside the distribution of the training set. Semi-supervised approaches enable scalable training by incorporating unlabeled data in a teacher-student pipeline. But still we need a large-scale supervision to have a reliable teacher model. The quality and scalability of labeled data are still the most critical part and this can be a possible bottleneck of generalizable representation learning research. For that reason, recent trend in representation learning is large-scale self-supervised learning. As we already discussed in Section 2.3, self-supervised learning approaches have been actively explored in computer vision and natural language processing. Especially, BERT [30] has changed the entire paradigm of natural language processing research.

Self-supervised representation learning of music has been mainly explored for generative models [34, 143, 32]. They are auto-regressive models that are trained to generate the next time steps. However, a recent work [85] reported that the transferred representations from Jukebox [32] are very effective in discriminative MIR tasks such as genre classification and key detection. Strictly speaking, Jukebox is not an unsupervised model because it conditions the model with styles and artists when it generates music. But the important take away from this is that the large-scale representation learning is effective in classification tasks. This supports the idea that we need to explore self-supervised music representation learning more.

Since transformers are reporting consistent success in many domains, not only in sequence modeling (e.g., text, speech) but also in non-sequential data (e.g., image), self-supervised research in music [143, 32] also has adopted the powerful architecture. Then one critical design choice is "how to extract a token-like representation of music". One can extract the short-

time audio representation using CNN [113], pretrain vector-quantized representations [32], or transcribe the audio to MIDI [143]. There is no standard in this short-time audio representation learning. We don't know which architecture works the best, even we don't know which temporal resolution is ideal for the transformer input. Also, each token (word) is semantically meaningful in NLP while a short-time music audio excerpt includes less information. Although the transformer's representation power guarantees the performance gain, these design choices need to be reviewed carefully.

6.3.2 Multimodality

Our music perception is multimodal. We listen to the audio, lyric adds another modality, and the music harmonizes with cover arts and music video. Music evokes emotion and it plays an important role in film making, storytelling, and various events. In this dissertation, we tried to bridge music with natural language semantics so that we can assist content creators to match suitable music to their stories. However, this can be extended towards all different modalities that are relevant to our music perception. We can bridge music with an actor's speech (text / audio) and facial expression (video), we can generate a photo album (image) with appropriate music, also we can assist video contents creators to discover background music. This does not limit to commercial music but also production music and license-free music. Different from commercial (popular) music, production music and license-free music do not have enough user-item interaction to use collaborative filtering, hence content filtering is critical. In this case, the multimodal embedding spaces can assist users to explore the catalog more efficiently.

Recently, a general framework for self-supervised learning in speech, vision and language (data2vec [221]) has been introduced. If we can incorporate all relevant data and modalities in multimodal music representation learning, each modality can supplement and compliment another to learn more robust representation. In consequence, the learned representation can get closer to human music perception and can enable more

versatile music retrieval applications.

6.3.3 Music in natural language

In Chapter 5, we demonstrated that natural language and music semantics can be bridged together to form a multimodal embedding space. We used sentences and paragraphs from books [213] or descriptions of certain antecedents [214] for the experiment. Based on the multimodal research, if we can collect descriptive text of the music content, it can facilitate natural language interface for music retrieval. For example, one can ask "play 80s disco tune with soulful vocal and danceable rhythm". Current systems need to parse the music tags and filter out based on the existing tag taxonomy. However, we can tackle the problem in an end-to-end fashion. To this end, there is a dataset such as MuMu dataset [222], or other descriptive text (album reviews) can be collected from the web. However, most descriptions exist in an album-level (multiple instance problem) and sometimes they don't directly talk about the audio content. Hence we need to create a dataset that directly describes the music audio content. A previous work [223] has shown that we can automatically generate music captions through the multimodal approach although the dataset is private. By creating a music caption dataset, we can move one step closer to human information resources at record stores.

Appendix A

LIST OF CONTRIBUTION

Tutorial / Online book

• Minz Won, Janne Spijkervet, and Keunwoo Choi. Music Classification: Beyond Supervised Learning, Towards Real-world Applications [224]
— International Society for Music Information Retrieval (ISMIR) 2021

Conference papers

- Minz Won, Justin Salamon, Nicholas J Bryan, Gautham J Mysore, and Xavier Serra. Emotion Embedding Spaces for Matching Music to Stories [225]
— International Society for Music Information Retrieval (ISMIR) 2021, **Best Student Paper**

- Minz Won, Keunwoo Choi, and Xavier Serra. Semi-supervised Music Tagging Transformer [113]
— International Society for Music Information Retrieval (ISMIR) 2021

- Wei-Tsung Lu, Ju-Chiang Wang, Minz Won, Keunwoo Choi, and Xuchen Song. SpecTNT: A Time-Frequency Transformer for Mu-

sic Audio [226]
— International Society for Music Information Retrieval (ISMIR) 2021

- Minz Won, Sergio Oramas, Oriol Nieto, Fabien Gouyon, and Xavier Serra. Multimodal Metric Learning for Tag-based Music Retrieval [227]
— IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2021

- Filip Korzeniowski, Oriol Nieto, Matthew McCallum, Minz Won, Sergio Oramas, and Erik Schmidt. Mood Classification Using Listening Data [175]
— International Society for Music Information Retrieval (ISMIR) 2020

- Minz Won, Sanghyuk Chun, Oriol Nieto, and Xavier Serra. Data-driven Harmonic Filters for Audio Representation Learning [108]
— IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2020

- Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. Evaluation of CNN-based Automatic Music Tagging Models [104]
— Sound and Music Computing (SMC) 2020

Workshop / Challenges / ArXiv

- Minz Won, Sanghyuk Chun, and Xavier Serra. Visualizing and Understanding Self-attention based Music Tagging — Machine Learning for Music Discovery Workshop, International Conference of Machine Learning (ICML) 2019

- Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The MTG-Jamendo dataset for automatic music tagging
— Machine Learning for Music Discovery Workshop, International Conference of Machine Learning (ICML) 2019

- Minz Won, Sanghyuk Chun, Oriol Nieto, and Xavier Serra. Automatic Music Tagging with Harmonic CNN
  — Late Break Demo, International Society for Music Information Retrieval (ISMIR) 2019

- Dmitry Bogdanov, Alastair Porter, Philip Tovstogan, and Minz Won. MediaEval 2019: Emotion and theme recognition in music using Jamendo
  — MeidaEval 2019, Challenge organizer

- Minz Won, Sanghyuk Chun, and Xavier Serra. Toward interpretable music tagging with self-attention
  — ArXiv 2019

- Jaehun Kim, Minz Won, Xavier Serra, and Cynthia CS Liem. Transfer learning of artist group factors to musical genre classification
  — The Web Conference (WWW) 2018, **Challenge winner**

Reviewer

- International Society of Music Information Retrieval (ISMIR)

- Sound and Music Computing (SMC)

Industrial contribution

- Research Internship at Kakao Corp.

- Research Internship at Naver Corp.

- Research Internship at Pandora Media Inc.

- Research Collaboration with Adobe

- Research Internship at ByteDance

Open source dataset

- MTG-Jamendo dataset https://github.com/MTG/mtg-jamendo-dataset

Open source code

- Emotion Embedding Spaces for Matching Music to Stories
https://github.com/minzwon/text2music-emotion-embedding

- Semi-supervised Music Tagging Transformer
https://github.com/minzwon/semi-supervised-music-tagging-transformer

- Multimodal Metric Learning for Tag-based Music Retrieval
https://github.com/minzwon/tag-based-music-retrieval

- Data-driven Harmonic Filters for Audio Representation Learning
https://github.com/minzwon/data-driven-harmonic-filters

- Evaluation of CNN-based Automatic Music Tagging Models
https://github.com/minzwon/sota-music-tagging-models

- Toward interpretable music tagging with self-attention
https://github.com/minzwon/self-attention-music-tagging

- Transfer learning of artist group factors to musical genre classification
https://gitlab.crowdai.org/minzwon/WWWMusicalGenreRecognitionChallenge

Bibliography

[1] Karp RM, Rabin MO. Efficient randomized pattern-matching algorithms. IBM journal of research and development. 1987;31(2):249–260.

[2] Knuth DE, Morris JH Jr, Pratt VR. Fast pattern matching in strings. SIAM journal on computing. 1977;6(2):323–350.

[3] Sunday DM. A very fast substring search algorithm. Communications of the ACM. 1990;33(8):132–142.

[4] Castelluccio M. The music genome project. Strategic Finance. 2006;p. 57–59.

[5] Mandel MI, Ellis DP. A web-based game for collecting music metadata. Journal of New Music Research. 2008;37(2):151–165.

[6] Corthaut N, Govaerts S, Verbert K, Duval E. Connecting the Dots: Music Metadata Generation, Schemas and Applications. In: Conference of the International Society for Music Information Retrieval (ISMIR); 2008. p. 249–254.

[7] Tzanetakis G, Cook P. Musical genre classification of audio signals. IEEE Transactions on speech and audio processing. 2002;10(5):293–302.

[8] Kim YE, Schmidt EM, Migneco R, Morton BG, Richardson P, Scott J, et al. Music emotion recognition: A state of the art review.

In: Conference of the International Society for Music Information Retrieval (ISMIR);. .

[9] Herrera-Boyer P, Peeters G, Dubnov S. Automatic classification of musical instrument sounds. Journal of New Music Research. 2003;32(1):3–21.

[10] Lamere P. Social tagging and music information retrieval. Journal of new music research. 2008;37(2):101–114.

[11] Haitsma J, Kalker T. A highly robust audio fingerprinting system. In: Conference of the International Society for Music Information Retrieval (ISMIR);. .

[12] Wang A, et al. An industrial strength audio search algorithm. In: Conference of the International Society for Music Information Retrieval (ISMIR);. .

[13] Ghias A, Logan J, Chamberlin D, Smith BC. Query by humming: Musical information retrieval in an audio database. In: ACM international conference on Multimedia;. .

[14] Serra J, Gómez E, Herrera P. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In: Advances in music information retrieval. Springer; 2010. p. 307–332.

[15] Yesiler F, Serrà J, Gómez E. Accurate and scalable version identification using musically-motivated embeddings. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2020. p. 21–25.

[16] Su X, Khoshgoftaar TM. A survey of collaborative filtering techniques. Advances in artificial intelligence. 2009;2009.

[17] Celma O. Music recommendation. In: Music recommendation and discovery. Springer; 2010. p. 43–85.

[18] Van Den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. In: Neural Information Processing Systems Conference (NIPS);. .

[19] Aucouturier JJ, Pachet F, et al. Music similarity measures: What's the use? In: Conference of the International Society for Music Information Retrieval (ISMIR); 2002. p. 13–17.

[20] Pampalk E, Flexer A, Widmer G, et al. Improvements of Audio-Based Music Similarity and Genre Classificaton. In: Conference of the International Society for Music Information Retrieval (ISMIR);. .

[21] Pujara J, Miao H, Getoor L, Cohen W. Knowledge graph identification. In: International Semantic Web Conference. Springer; 2013. p. 542–557.

[22] Oramas S, Ostuni VC, Noia TD, Serra X, Sciascio ED. Sound and music recommendation with knowledge graphs. ACM Transactions on Intelligent Systems and Technology (TIST). 2016;8(2):1–21.

[23] Korzeniowski F, Oramas S, Gouyon F. Artist Similarity with Graph Neural Networks. Conference of the International Society for Music Information Retrieval (ISMIR). 2021;.

[24] Sturm BL. A simple method to determine if a music information retrieval system is a "horse". IEEE Transactions on Multimedia. 2014;16(6):1636–1644.

[25] Kereliuk C, Sturm BL, Larsen J. Deep learning and music adversaries. IEEE Transactions on Multimedia. 2015;17(11):2059–2071.

[26] Ganchev T, Fakotakis N, Kokkinakis G. Comparative evaluation of various MFCC implementations on the speaker verification task. In: Proceedings of the SPECOM. vol. 1; 2005. p. 191–194.

[27] Fujishima T. Real-time chord recognition of musical sound: A system using common lisp music. Proc ICMC, Oct 1999. 1999;p. 464–467.

[28] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence. 2013;35(8):1798–1828.

[29] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems (NIPS). 2012;25:1097–1105.

[30] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1; 2019. .

[31] Engel J, Hantrakul LH, Gu C, Roberts A. DDSP: Differentiable Digital Signal Processing. In: International Conference on Learning Representations (ICLR); 2020. .

[32] Dhariwal P, Jun H, Payne C, Kim JW, Radford A, Sutskever I. Jukebox: A generative model for music. arXiv preprint arXiv:200500341. 2020;.

[33] Choi K, Fazekas G, Sandler M. Automatic tagging using deep convolutional neural networks. In: Proc. of International Society for Music Information Retrieval Conference (ISMIR); 2016. .

[34] Huang CZA, Vaswani A, Uszkoreit J, Shazeer N, Simon I, Hawthorne C, et al. Music transformer. In: International Conference on Learning Representations (ICLR); 2019. .

[35] Pacha A, Hajič J, Calvo-Zaragoza J. A baseline for general music object detection with deep learning. Applied Sciences. 2018;8(9):1488.

[36] Brendel W, Bethge M. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In: International Conference on Learning Representations (ICLR); 2019. .

[37] Dietterich TG, Lathrop RH, Lozano-Pérez T. Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence. 1997;89(1-2):31–71.

[38] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems (NIPS); 2017. p. 5998–6008.

[39] Park J, Lee J, Park J, Ha JW, Nam J. Representation learning of music using artist labels. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2018. .

[40] Bogdanov D, Won M, Tovstogan P, Porter A, Serra X. The MTG-Jamendo dataset for automatic music tagging. Machine Learning for Music Discovery Workshop, International Conference on Machine Learning. 2019;.

[41] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: International Conference on Learning Representations (ICLR); 2013. .

[42] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proc. of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. .

[43] Bertin-Mahieux T, Ellis DP, Whitman B, Lamere P. The million song dataset. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2011. .

[44] Gouyon F, Klapuri A, Dixon S, Alonso M, Tzanetakis G, Uhle C, et al. An experimental comparison of audio tempo induction

algorithms. IEEE Transactions on Audio, Speech, and Language Processing. 2006;14(5):1832–1844.

[45] Defferrard M, Benzi K, Vandergheynst P, Bresson X. FMA: A dataset for music analysis. In: Conference of the International Society for Music Information Retrieval (ISMIR); 2017. .

[46] Cano P, Gómez E, Gouyon F, Herrera P, Koppenberger M, Ong B, et al. ISMIR 2004 audio description contest. Music Technology Group of the Universitat Pompeu Fabra, Tech Rep. 2006;.

[47] Schreiber H. Improving Genre Annotations for the Million Song Dataset. In: Conference of the International Society for Music Information Retrieval (ISMIR); 2015. p. 241–247.

[48] Bogdanov D, Porter A, Schreiber H, Urbano J, Oramas S. The acousticbrainz genre dataset: Multi-source, multi-level, multi-label, and large-scale. In: Conference of the International Society for Music Information Retrieval (ISMIR); 2019. .

[49] Koelstra S, Muhl C, Soleymani M, Lee JS, Yazdani A, Ebrahimi T, et al. Deap: A database for emotion analysis; using physiological signals. IEEE transactions on affective computing. 2011;3(1):18–31.

[50] Soleymani M, Caro MN, Schmidt EM, Sha CY, Yang YH. 1000 songs for emotional analysis of music. In: Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia; 2013. p. 1–6.

[51] Soleymani M, Aljanaki A, Yang Y. DEAM: MediaEval database for emotional analysis in Music. Geneva, Switzerland; 2016.

[52] Bosch JJ, Janer J, Fuhrmann F, Herrera P. A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2012. p. 559–564.

[53] Humphrey E, Durand S, McFee B. OpenMIC-2018: An Open Data-set for Multiple Instrument Recognition. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2018. p. 438–444.

[54] Law E, West K, Mandel MI, Bay M, Downie JS. Evaluation of algorithms using games: The case of music tagging. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2009. .

[55] Bertin-Mahieux T, Eck D, Mandel M. Automatic tagging of audio: The state-of-the-art. In: Machine audition: Principles, algorithms and systems. IGI Global; 2011. p. 334–352.

[56] Dieleman S, Schrauwen B. End-to-end learning for music audio. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2014. p. 6964–6968.

[57] Lee J, Park J, Kim KL, Nam J. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. In: Sound and music computing (SMC); 2017. .

[58] Kim T, Lee J, Nam J. Sample-level CNN architectures for music auto-tagging using raw waveforms. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2018. p. 366–370.

[59] Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. In: International conference on Machine learning (ICML); 2006. p. 233–240.

[60] McFee B, Salamon J, Bello JP. Adaptive pooling operators for weakly labeled sound event detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2018;26(11):2180–2193.

[61] Pachet F. Knowledge management and musical metadata. Idea Group. 2005;12.

[62] Cohen WW, Fan W. Web-collaborative filtering: Recommending music by crawling the web. Computer Networks. 2000;33(1-6):685–698.

[63] Torrey L, Shavlik J. Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global; 2010. p. 242–264.

[64] Choi K, Fazekas G, Sandler M, Cho K. Transfer learning for music classification and regression tasks. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2017. .

[65] Kim J, Won M, Serra X, Liem CC. Transfer learning of artist group factors to musical genre classification. In: Companion Proceedings of The Web Conference 2018; 2018. p. 1929–1934.

[66] Defferrard M, Mohanty SP, Carroll SF, Salathé M. Learning to recognize musical genre from audio. The Web Conference 2018. 2018;.

[67] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining. Ieee; 2008. p. 263–272.

[68] Yalniz IZ, Jégou H, Chen K, Paluri M, Mahajan D. Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:190500546. 2019;.

[69] Simard PY, Steinkraus D, Platt JC, et al. Best practices for convolutional neural networks applied to visual document analysis. In: Icdar. vol. 3; 2003. .

[70] Grandvalet Y, Bengio Y, et al. Semi-supervised learning by entropy minimization. CAP. 2005;367:281–296.

[71] Lee DH, et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3; 2013. p. 896.

[72] Zhu X, Ghahramani Z, Lafferty JD. Semi-supervised learning using gaussian fields and harmonic functions. In: International conference on Machine learning (ICML); 2003. p. 912–919.

[73] Kingma DP, Mohamed S, Rezende DJ, Welling M. Semi-supervised learning with deep generative models. In: Advances in neural information processing systems (NIPS); 2014. p. 3581–3589.

[74] Berthelot D, Carlini N, Goodfellow I, Papernot N, Oliver A, Raffel C. Mixmatch: A holistic approach to semi-supervised learning. Conference on Neural Information Processing Systems (NeurIPS). 2019;.

[75] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond empirical risk minimization. International Conference on Learning Representations (ICLR). 2018;.

[76] Xie Q, Luong MT, Hovy E, Le QV. Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 10687–10698.

[77] He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 9729–9738.

[78] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: International conference on machine learning. PMLR; 2020. p. 1597–1607.

[79] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

[80] Spijkervet J, Burgoyne JA. Contrastive Learning of Musical Representations. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2021. .

[81] Oord Avd, Li Y, Vinyals O. Representation learning with contrastive predictive coding. In: Advances in neural information processing systems (NIPS); 2018. .

[82] Schneider S, Baevski A, Collobert R, Auli M. wav2vec: Unsupervised pre-training for speech recognition. In: Interspeech; 2019. .

[83] Baevski A, Zhou H, Mohamed A, Auli M. wav2vec 2.0: A framework for self-supervised learning of speech representations. In: Advances in Neural Information Processing Systems (NeurIPS); 2020. .

[84] Oord Avd, Vinyals O, Kavukcuoglu K. Neural discrete representation learning. In: Advances in Neural Information Processing Systems (NIPS); 2017. .

[85] Castellon R, Donahue C, Liang P. Codified audio language modeling learns useful representations for music information retrieval. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2021. .

[86] Pascual S, Ravanelli M, Serra J, Bonafonte A, Bengio Y. Learning problem-agnostic speech representations from multiple self-supervised tasks. In: Interspeech; 2019. .

[87] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: Transformers

for image recognition at scale. In: International Conference on Learning Representations (ICLR); 2021. .

[88] Arnab A, Dehghani M, Heigold G, Sun C, Lučić M, Schmid C. Vivit: A video vision transformer. In: International Conference on Computer Vision (ICCV); 2021. .

[89] Gong Y, Lai CIJ, Chung YA, Glass J. SSAST: Self-Supervised Audio Spectrogram Transformer. In: AAAI Conference on Artificial Intelligence; 2021. .

[90] Choi J, Lee J, Park J, Nam J. Zero-shot learning for audio-based music classification and tagging. Conference of the International Society for Music Information Retrieval Conference (ISMIR). 2019;.

[91] Won M, Oramas S, Nieto O, Gouyon F, Serra X. Multimodal Metric Learning for Tag-based Music Retrieval. In Proc of International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2021;.

[92] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735–1780.

[93] Sak H, Senior AW, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014;.

[94] Li X, Wu X. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2015. p. 4520–4524.

[95] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: Advances in neural information processing systems (NIPS); 2014. p. 3104–3112.

[96] Choi K, Fazekas G, Sandler M, Cho K. Convolutional recurrent neural networks for music classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017. p. 2392–2396.

[97] Grill JB, Strub F, Altché F, Tallec C, Richemond PH, Buchatskaya E, et al. Bootstrap your own latent: A new approach to self-supervised learning. In: Advances in Neural Information Processing Systems (NeurIPS); 2020. .

[98] Parmar N, Vaswani A, Uszkoreit J, Kaiser L, Shazeer N, Ku A, et al. Image transformer. In: International Conference on Machine Learning (ICML). PMLR; 2018. .

[99] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. In: Advances in Neural Information Processing Systems (NeurIPS); 2020. .

[100] MatthewDavies E, Böck S. Temporal convolutional networks for musical audio beat tracking. In: Proc. of European Signal Processing Conference (EUSIPCO); 2019. .

[101] Kim JW, Salamon J, Li P, Bello JP. Crepe: A convolutional representation for pitch estimation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2018. p. 161–165.

[102] Gururani S, Sharma M, Lerch A. An attention mechanism for musical instrument recognition. In: Conference of International Society for Music Information Retrieval (ISMIR); 2019. .

[103] Delbouys R, Hennequin R, Piccoli F, Royo-Letelier J, Moussallam M. Music mood detection based on audio and lyrics with deep neural net. In: Conference of International Society for Music Information Retrieval (ISMIR); 2018. .

[104] Won M, Ferraro A, Bogdanov D, Serra X. Evaluation of cnn-based automatic music tagging models. In: Sound and Music Computing (SMC); 2020. .

[105] Choi K, Fazekas G, Cho K, Sandler M. The effects of noisy labels on deep convolutional neural networks for music classification. IEEE Transactions on Emerging Topics in Computational Intelligence. 2018;.

[106] McFee B, Humphrey EJ, Bello JP. A software framework for musical data augmentation. In: Conference of International Society for Music Information Retrieval (ISMIR);. .

[107] Pons J, Nieto O, Prockup M, Schmidt E, Ehmann A, Serra X. End-to-end learning for music audio tagging at scale. In: Proc. of the 19th International Society for Music Information Retrieval Conference (ISMIR); 2018. .

[108] Won M, Chun S, , Nieto O, Serra X. Data-driven harmonic filters for audio representation learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2020. .

[109] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR); 2015. .

[110] Heo B, Chun S, Oh SJ, Han D, Yun S, Kim G, et al. Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights. In: International Conference on Learning Representations (ICLR); 2021. .

[111] Loshchilov I, Hutter F. Decoupled weight decay regularization. In: International Conference on Learning Representations (ICLR); 2019. .

[112] Pons J, Nieto O, Prockup M, Schmidt E, Ehmann A, Serra X. End-to-end learning for music audio tagging at scale. In: Conference of the International Society for Music Information Retrieval (ISMIR); 2018. .

[113] Won M, Choi K, Serra X. Semi-supervised music tagging transformer. In: Conference of International Society for Music Information Retrieval (ISMIR); 2021. .

[114] Won M, Chun S, Serra X. Toward interpretable music tagging with self-attention. arXiv preprint arXiv:190604972. 2019;.

[115] Choi K, Fazekas G, Sandler M. Automatic tagging using deep convolutional neural networks. Conference of the International Society for Music Information Retrieval Conference (ISMIR). 2016;.

[116] Bittner RM, McFee B, Salamon J, Li P, Bello JP. Deep Salience Representations for F0 Estimation in Polyphonic Music. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2017. p. 63–70.

[117] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proc. of the IEEE conference on computer vision and pattern recognition (CVPR); 2018. p. 7132–7141.

[118] Dolby E. Standards and practices for authoring Dolby Digital and Dolby E bitstreams. Dolby Labortories, Inc. 2002;.

[119] Hannun A, Case C, Casper J, Catanzaro B, Diamos G, Elsen E, et al. Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:14125567. 2014;.

[120] Sainath TN, Weiss RJ, Senior A, Wilson KW, Vinyals O. Learning the speech front-end with raw waveform CLDNNs. In: the 16th Annual Conference of the International Speech Communication Association; 2015. .

[121] Dai W, Dai C, Qu S, Li J, Das S. Very deep convolutional neural networks for raw waveforms. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017. .

[122] Sethares WA. Tuning, timbre, spectrum, scale. Springer Science & Business Media; 2005.

[123] Venkataramani S, Casebeer J, Smaragdis P. Adaptive front-ends for end-to-end source separation. In: the 31st Conference on Neural Information Processing Systems; 2017. .

[124] Seki H, Yamamoto K, Nakagawa S. A deep neural network integrated with filterbank learning for speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017. .

[125] Dörfler M, Grill T, Bammer R, Flexer A. Basic filters for convolutional neural networks applied to music: Training or design? Neural Computing and Applications. 2018;.

[126] Takeuchi D, Yatabe K, Koizumi Y, Oikawa Y, Harada N. Data-driven design of perfect reconstruction filterbank for DNN-based sound source enhancement. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2019. .

[127] Ravanelli M, Bengio Y. Speaker recognition from raw waveform with sincnet. In: IEEE Spoken Language Technology Workshop; 2018. .

[128] Won M, Chun S, Nieto O, Serra X. Automatic music tagging with harmonic CNN. In: Late Breaking Demo in the International Society for Music Information Retrieval Conference; 2019. .

[129] Glasberg BR, Moore BC. Derivation of auditory filter shapes from notched-noise data. Hearing research. 1990;47(1-2):103–138.

[130] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. the 32nd International Conference on Machine Learning. 2015;.

[131] de Andrade DC, Leo S, Viana MLDS, Bernkopf C. A neural attention model for speech command recognition. arXiv preprint arXiv:180808929. 2018;.

[132] Xu Y, Kong Q, Wang W, Plumbley MD. Surrey-CVSSP system for DCASE2017 challenge task4. Detection and Classification of Acoustic Scenes and Events. 2017;.

[133] Warden P. Speech commands: A dataset for limited-vocabulary speech recognition. arXiv preprint arXiv:180403209. 2018;.

[134] Mesaros A, Heittola T, Diment A, Elizalde B, Shah A, Vincent E, et al. DCASE 2017 challenge setup: Tasks, datasets and baseline system. In: Detection and Classification of Acoustic Scenes and Events; 2017. .

[135] Gemmeke JF, Ellis DP, Freedman D, Jansen A, Lawrence W, Moore RC, et al. Audio set: An ontology and human-labeled dataset for audio events. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017. .

[136] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. In: Proceedings of the International Conference on Machine learning (ICML); 2013. .

[137] Choi K, Fazekas G, Sandler M. Explaining deep convolutional neural networks on music classification. arXiv preprint arXiv:160702444. 2016;.

[138] Mishra S, Sturm BL, Dixon S. Local interpretable model-agnostic explanations for music content analysis. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2017. .

[139] Mishra S, Sturm BL, Dixon S. Understanding a deep machine listening model through feature inversion. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2018. .

[140] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. Technical report, OpenAi; 2018.

[141] Zhang H, Goodfellow I, Metaxas D, Odena A. Self-attention generative adversarial networks. Proceedings of the International Conference on Machine learning (ICML). 2019;.

[142] Parmar N, Vaswani A, Uszkoreit J, Kaiser Ł, Shazeer N, Ku A, et al. Image transformer. Proceedings of the International Conference on Machine learning (ICML). 2018;.

[143] Hawthorne C, Stasyuk A, Roberts A, Simon I, Huang CZA, Dieleman S, et al. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In: International Conference on Learning Representations (ICLR); 2019. .

[144] Hawthorne C, Elsen E, Song J, Roberts A, Simon I, Raffel C, et al. Onsets and frames: Dual-objective piano transcription. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2018. .

[145] Van Den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, et al. WaveNet: A generative model for raw audio. Speech Synthesis Workshop (SSW). 2016;.

[146] Hoffer E, Hubara I, Soudry D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In: Proceedings of the Advances in Neural Information Processing Systems (NIPS); 2017. .

[147] Seong S, Lee Y, Kee Y, Han D, Kim J. Towards Flatter Loss Surface via Nonmonotonic Learning Rate Scheduling. In: Conference on Uncertainty in Artificial Intelligence (UAI); 2018. .

[148] Keskar NS, Socher R. Improving generalization performance by switching from adam to sgd. arXiv preprint arXiv:171207628. 2017;.

[149] Wilson AC, Roelofs R, Stern M, Srebro N, Recht B. The marginal value of adaptive gradient methods in machine learning. In: Proceedings of the Advances in Neural Information Processing Systems (NIPS); 2017. .

[150] Bogdanov D, Wack N, Gómez Gutiérrez E, Gulati S, Herrera Boyer P, Mayor O, et al. Essentia: An audio analysis library for music information retrieval. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2013. .

[151] McFee B, Raffel C, Liang D, Ellis DP, McVicar M, Battenberg E, et al. librosa: Audio and music signal analysis in python. In: Proceedings of the python in science conference; 2015. .

[152] Lee J, Shin JH, Kim JS. Interactive visualization and manipulation of attention-based neural machine translation. In: Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations; 2017. .

[153] Spijkervet J. Spijkervet/torchaudio-augmentations. Zenodo; 2021. Available from: https://doi.org/10.5281/zenodo.5042440.

[154] Lloyd S. Least squares quantization in PCM. IEEE transactions on information theory. 1982;28(2):129–137.

[155] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of machine Learning research. 2003;3(Jan):993–1022.

[156] Herrera P, Streich S. Detrended Fluctuation Analysis of Music Signals: Danceability Estimation and further Semantic Characterization. In: Audio Engineering Society Convention 118. Audio Engineering Society; 2005. .

[157] Chapelle O, Scholkopf B, Zien A. Semi-supervised learning. IEEE Transactions on Neural Networks. 2009;20(3):542–542.

[158] Chen T, Kornblith S, Swersky K, Norouzi M, Hinton G. Big self-supervised models are strong semi-supervised learners. In: Conference on Neural Information Processing Systems (NeurIPS); 2020. .

[159] Li J, Seltzer ML, Wang X, Zhao R, Gong Y. Large-scale domain adaptation via teacher-student learning. In: Interspeech; 2017. .

[160] Kum S, Lin JH, Su L, Nam J. Semi-supervised learning using teacher-student models for vocal melody extraction. In: Conference of International Society for Music Information Retrieval (ISMIR); 2020. .

[161] Kim Y, Rush AM. Sequence-level knowledge distillation. Proc of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2016;.

[162] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. Advances in neural information processing systems (NIPS), Deep learning workshop. 2015;.

[163] Guo W, Wang J, Wang S. Deep multimodal representation learning: A survey. IEEE Access. 2019;7:63373–63394.

[164] Xing EP, Jordan MI, Russell SJ, Ng AY. Distance metric learning with application to clustering with side-information. In: Advances in neural information processing systems; 2003. .

[165] Weinberger KQ, Saul LK. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research. 2009;10(2).

[166] Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop; 2015. .

[167] Wang L, Li Y, Lazebnik S. Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. .

[168] Oramas S, Barbieri F, Nieto O, Serra X. Multimodal deep learning for music genre classification. Transactions of the International Society for Music Information Retrieval 2018; 1 (1): 4-21. 2018;.

[169] Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Ranzato M, et al. Devise: A deep visual-semantic embedding model. In: Advances in neural information processing systems (NIPS); 2013. .

[170] Surís D, Duarte A, Salvador A, Torres J, Giró-i Nieto X. Cross-modal embeddings for video and audio retrieval. In: Proc. of the European Conference on Computer Vision; 2018. .

[171] Hoffer E, Ailon N. Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition. Springer; 2015. .

[172] Herlocker JL, Konstan JA, Riedl J. Explaining collaborative filtering recommendations. In: Proc. of ACM conference on Computer supported cooperative work; 2000. .

[173] Hu Y, Koren Y, Volinsky C. Collaborative Filtering for Implicit Feedback Datasets. In: Proc. of the 8th IEEE International Conference on Data Mining; 2008. .

[174] McFee B, Bertin-Mahieux T, Ellis DPW, Lanckriet GRG. The million song dataset challenge. Proc of the 21st international conference companion on World Wide Web;.

[175] Korzeniowski F, Nieto O, McCallum M, Won M, Oramas S, Schmidt E. Mood classification using listening data. In: Conference of International Society for Music Information Retrieval (ISMIR); 2020. .

[176] Wu CY, Manmatha R, Smola AJ, Krahenbuhl P. Sampling matters in deep embedding learning. In: Proc. of the IEEE International Conference on Computer Vision; 2017. .

[177] Simo-Serra E, Trulls E, Ferraz L, Kokkinos I, Fua P, Moreno-Noguer F. Discriminative learning of deep convolutional feature point descriptors. In: Proc. of the IEEE International Conference on Computer Vision; 2015. .

[178] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In: Proc. of the IEEE conference on computer vision and pattern recognition; 2015. .

[179] Wang B, Yang Y, Xu X, Hanjalic A, Shen HT. Adversarial cross-modal retrieval. In: Proc. of the 25th ACM International Conference on Multimedia; 2017. .

[180] Yao T, Mei T, Ngo CW. Learning query and image similarities with ranking canonical correlation analysis. In: Proc. of the IEEE International Conference on Computer Vision (ICCV); 2015. .

[181] Zhen L, Hu P, Wang X, Peng D. Deep supervised cross-modal retrieval. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. .

[182] Wei Y, Zhao Y, Lu C, Wei S, Liu L, Zhu Z, et al. Cross-modal retrieval with CNN visual features: A new baseline. IEEE Transactions on Cybernetics. 2016;47(2).

[183] Wang K, Yin Q, Wang W, Wu S, Wang L. A comprehensive survey on cross-modal retrieval. arXiv preprint arXiv:160706215. 2016;.

[184] Mohammad SM, Turney PD. Crowdsourcing a word–emotion association lexicon. Computational Intelligence. 2013;29(3).

[185] Strapparava C, Valitutti A, et al. Wordnet affect: an affective extension of wordnet. In: Proc. of International Conference on Language Resources and Evaluation; 2004. .

[186] Danisman T, Alpkocak A. Feeler: Emotion classification of text using vector space model. In: AISB Convention: Communication, Interaction and Social Intelligence. vol. 1; 2008. .

[187] Hasan M, Rundensteiner E, Agu E. Emotex: Detecting emotions in twitter messages. 2014;.

[188] Abdul-Mageed M, Ungar L. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In: Proc. of annual meeting of the association for computational linguistics; 2017. .

[189] Batbaatar E, Li M, Ryu KH. Semantic-emotion neural network for emotion recognition from text. IEEE Access. 2019;7.

[190] Cortiz D. Exploring Transformers in Emotion Recognition: a comparison of BERT, DistillBERT, RoBERTa, XLNet and ELECTRA. arXiv preprint arXiv:210402041. 2021;.

[191] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692. 2019;.

[192] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Neural Information Processing Systems Workshop on Energy Efficient Machine Learning and Cognitive Computing. 2019;.

[193] Tzanetakis G. Marsyas submissions to MIREX 2007. Music Information Retrieval Evaluation eXchange. 2007;.

[194] Peeters G. A generic training and classification system for MIREX08 classification tasks: audio music mood, audio genre, audio artist and audio tag. In: Conference of International Society for Music Information Retrieval (ISMIR); 2008. .

[195] Cao C, Li M. Thinkit's submissions for MIREX2009 audio music classification and similarity tasks. Music Information Retrieval Evaluation eXchange. 2009;.

[196] Lidy T, Schindler A, et al. Parallel convolutional neural networks for music genre and mood classification. Music Information Retrieval Evaluation eXchange. 2016;.

[197] Lee J, Bryan NJ, Salamon J, Jin Z, Nam J. Metric Learning VS Classification for Disentangled Music Representation Learning. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2020. .

[198] Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proc. of the IEEE International Conference on Computer Vision (ICCV); 2017. .

[199] MediaEval 2020 Emotion and Theme Recognition in Music Task: Loss Function Approaches for Multi-label Music Tagging. MediaEval2020. 2020;.

[200] Schmidt EM, Turnbull D, Kim YE. Feature selection for content-based, time-varying musical emotion regression. In: International conference on Multimedia information retrieval; 2010. p. 267–274.

[201] Han Bj, Rho S, Dannenberg RB, Hwang E. SMERS: Music Emotion Recognition Using Support Vector Regression. In: Conference of the International Society for Music Information Retrieval Conference (ISMIR); 2009. .

[202] Russell JA. A circumplex model of affect. Journal of personality and social psychology. 1980;39(6):1161.

[203] Thayer RE. The biopsychology of mood and arousal. Oxford University Press; 1990.

[204] Mohammad S. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2018. p. 174–184.

[205] Doh S, Lee J, Park TH, Nam J. Musical Word Embedding: Bridging the Gap between Listening Contexts and Music. Machine Learning for Media Discovery Workshop, International Conference on Machine Learning (ICML). 2020;.

[206] Arandjelovic R, Zisserman A. Look, listen and learn. In: Proc. of the IEEE International Conference on Computer Vision (ICCV); 2017. p. 609–617.

[207] Cramer J, Wu HH, Salamon J, Bello JP. Look, listen, and learn more: Design choices for deep audio embeddings. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2019. .

[208] Huh J, Lee M, Heo H, Mun S, Chung JS. Metric learning for keyword spotting. In: 2021 IEEE Spoken Language Technology Workshop (SLT). IEEE; 2021. p. 133–140.

[209] Elizalde B, Zarar S, Raj B. Cross modal audio search and retrieval with joint embeddings based on text and audio. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2019. .

[210] Oncescu AM, Koepke A, Henriques JF, Akata Z, Albanie S. Audio Retrieval with Natural Language Queries. In: Interspeech; 2021. .

[211] Wang L, Li Y, Huang J, Lazebnik S. Learning two-branch neural networks for image-text matching tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018;41(2):394–407.

[212] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: State-of-the-Art Natural Language Processing. In: Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2020. .

[213] Alm ECO. Affect in* text and speech. Citeseer; 2008.

[214] Scherer KR, Wallbott HG. Evidence for universality and cultural variation of differential emotion response patterning. Journal of personality and social psychology. 1994;66(2):310.

[215] Çano E, Morisio M, et al. Music mood dataset creation based on last. fm tags. In: 2017 International Conference on Artificial Intelligence and Applications, Vienna, Austria; 2017. .

[216] Hu X, Downie JS, Ehmann AF. Lyric text mining in music mood classification. American music. 2009;183(5,049):2–209.

[217] McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. The Journal of Open Source Software. 2018;3(29):861.

[218] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(11).

[219] Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:181112231. 2018;.

[220] Sturm BL. The “horse” inside: Seeking causes behind the behaviors of music content analysis systems. Computers in Entertainment (CIE). 2017;14(2):1–32.

[221] Baevski A, Hsu WN, Xu Q, Babu A, Gu J, Auli M. Data2vec: A general framework for self-supervised learning in speech, vision and language. arXiv preprint arXiv:220203555. 2022;.

[222] Oramas S, Nieto O, Barbieri F, Serra X. Multi-label music genre classification from audio, text, and images using deep features. Conference of the International Society for Music Information Retrieval Conference (ISMIR). 2017;.

[223] Manco I, Benetos E, Quinton E, Fazekas G. MusCaps: Generating Captions for Music Audio. In: 2021 International Joint Conference on Neural Networks (IJCNN). IEEE; 2021. p. 1–8.

[224] Won M, Spijkervet J, Choi K. Music Classification: Beyond Supervised Learning, Towards Real-world Applications. International Society for Music Information Retrieval Conference (ISMIR) tutorial. 2021;.

[225] Won M, Salamon J, Bryan NJ, Mysore GJ, Serra X. Emotion Embedding Spaces for Matching Music to Stories. In: Conference of International Society for Music Information Retrieval (ISMIR); 2021. .

[226] Lu WT, Wang JC, Won M, Choi K, Song X. SpecTNT: a time-frequency transformer for music audio. In: Conference of International Society for Music Information Retrieval (ISMIR); 2021. .

[227] Won M, Oramas S, Nieto O, Gouyon F, Serra X. Multimodal metric learning for tag-based music retrieval. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2021. p. 591–595.