



Universiteit
Leiden

Master Computer Science

Multimodal Self-supervised Music Genre Classification with Audio and Lyrics

Name: Kaiteng Jiang
Student ID: s3479420
Date: 16/10/2024
Specialisation: Data Science: Computer Science
1st supervisor: Prof. dr. Holger H. Hoos
2nd supervisor: Dr. Igor Vatulkin

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Music genre classification is a fundamental task in the field of music information retrieval, playing a critical role in organising, recommending, and discovering music. Traditional approaches often rely on single-modal data, such as audio features, limiting the scope of information that can be used to effectively distinguish between genres. This thesis presented a multimodal approach to music genre classification, incorporating both audio and lyrics to capture a more comprehensive representation of music. By leveraging the potential complementarity of these two modalities, the proposed method improved classification performance by utilizing not only the acoustic characteristics but also the semantic content present in lyrics. To mitigate the scarcity of multimodal music datasets, this research employed a self-supervised learning framework. Specifically, contrastive learning and masked data modeling were combined to enable the model to learn both intramodal information and intermodal relationships from large amounts of unlabeled music data. The joint usage of two self-supervised learning methods resulted in a remarkable improvement of model performance. However, in comparison with other baseline models, the performance of our model remained constrained by the relatively small dataset size.

Table of Contents

List of Figures	iii
List of Tables	iv
1 Introduction	1
2 Background	3
2.1 Transformers	3
2.2 Self-supervised Learning	4
2.2.1 Pretext tasks	4
2.3 Multimodal Learning	7
2.3.1 Modality fusion and alignment	8
2.3.2 Multimodal self-supervised learning	9
3 Related Work	12
3.1 Self-supervised learning in MIR	12
3.2 Classifying music genres with audio and text	13
3.2.1 Music genre classification with audio	13
3.2.2 Music genre classification with lyrics	14
3.2.3 Multimodal music genre classification with audio and lyrics	14
4 Methods	16
4.1 Tokenizing audio and lyrics	16
4.2 Self-supervised pretraining	16
4.2.1 Audio-lyrics MAE	16
4.2.2 Contrastive audio-lyrics learning	18
4.3 Fine-tuning and classification	19
4.4 Motivation behind the methods	19
4.4.1 Why audio and lyrics	19
4.4.2 Combining two self-supervised learning methods	20
4.4.3 Modality fusion	20
5 Experiments	22
5.1 Datasets	22
5.2 Experimental settings	23
5.2.1 Hyperparameters	23
5.2.2 Metrics	24
5.3 Experimental design	25
6 Results	27
6.1 Ablation of modalities	27
6.2 Ablation of self-supervised learning methods	28
6.3 Hyperparameter tuning	29

7	Discussion	33
7.1	Conclusion	33
7.2	Limitations	34
7.3	Future research	34
	References	35

List of Figures

2.1	The performance of self-supervised learning methods on ImageNet top-1 accuracy (i.e, the conventional classification accuracy in which the prediction with the highest accuracy must be the exact ground truth.) in March 2021 [Liu+21b]. The horizontal axis was the model size, in terms of the number of parameters; the vertical axis was the accuracy, which reflected the model performance. The ability of many self-supervised contrastive learning methods (e.g., SimCLR, MoCo v2, BYOL, SwAV) were almost comparable to, or even surpassed the supervised learning method (ResNet 50).	5
2.2	The architecture of MAE. Images are patchified and a large proportion of patches (75%) are masked then reconstructed [He+22].	7
2.3	The categorization of modality alignment [ZMH23].	8
2.4	An example of multimodal contrastive learning from the paper of CAV-MAE [Gon+23].	10
2.5	An illustration of the multimodal MDM, where the model predicts the corrupted input of one modality based on the information of the other [ZMH23].	10
4.1	The tokenization of audio spectrograms by splitting them into non-overlapping square patches.	16
4.2	The model framework in the pretraining stage. The input sequences of audio and lyrics were masked and pass through the modal-specific encoders, and took a multi-stream data pass to the joint encoder to compute the contrastive loss and predict the masked tokens through a joint decoder.	17
4.3	The model framework in the fine-tuning stage. No masking operation was performed, and the decoder was replaced by a linear classifier.	19
5.1	The distribution of 120 genre labels of the highest frequency.	22
5.2	The distribution of the lyrics length before and after preprocessing. Both mean and variance decreased.	23
6.1	The modality importances of audio and lyrics each (coarse) genre. We also calculated the ratio of audio importance to lyrics importance.	27

List of Tables

6.1	Results of the modality ablation study. "A" referred to audio and "L" referred to lyrics. All the results were averaged on 5-fold cross validation, presented with mean and standard deviation values. The p-values of the statistical tests were marked by the number of asterisks: (*)- $p < 0.05$, (**) - $p < 0.01$, (***)- $p < 0.001$	27
6.2	Results of the self-supervised learning ablation study. The p-values of the statistical tests were marked by the number of asterisks: (*)- $p < 0.05$, (**) - $p < 0.01$, (***)- $p < 0.001$. (–) meanted that there was no statistical significance.	28
6.3	Results of comparing different audio lengths. We also recorded the highest contrastive accuracy in the pretraining stage, which was the ratio of cases in which a sample was more similar to its positive counterpart than any other negative ones. . .	29
6.4	Result of comparing different lyrics length in number of tokens. CL Acc was the highest contrastive accuracy recorded in the pretraining stage.	29
6.5	Results of using different pretraining datasets: pretraining on our original dataset m4a, and on m4a + msd together. The p-values of the statistical tests were marked by the number of asterisks: (*)- $p < 0.05$, (**) - $p < 0.01$, (***)- $p < 0.001$. (–) meanted that there was no statistical significance ($p > 0.05$).	30
6.6	Results of comparing different model sizes (numbers of hidden layers in each modality-specific encoder and the joint decoder).	30
6.7	Results of comparing different batch sizes in contrastive learning.	30
6.8	Results of comparing different masking ratios of (audio, lyrics). CL Acc was the highest contrastive accuracy recorded in the pretraining stage.	31
6.9	Results of comparing our model to three other baselines: the unimodal Inception v4 (audio) and BERT (lyrics), as well as the multimodal ALNet.	31
6.10	Results of audio spectrogram inpainting under different masking ratios: 0.5, 0.75 and 0.85.	32

Chapter 1

Introduction

The term "genre" in music refers to "a set of musical events whose course is governed by rules (of any kind) accepted by a community" [Fab+07], as defined by musicologist Franco Fabbri. Music genre classification [TC02; LOL03; MF06] is a challenging and fundamental problem in the field of music information retrieval (MIR) [Dow03; SGU+14]. The ability to automatically identify and categorize music by genre is critical in organizing vast digital music libraries, improving recommendation systems, and enhancing user experiences in streaming platforms [EA20; LOL03]. With the rapid growth of digital music databases, the development of robust and efficient genre classification systems has become increasingly important for both academic research and commercial applications. Despite its practical significance, the task remains complex due to the inherent overlap between genres, cultural influences, and the subjective nature of genre boundaries. Given the complexity of music genre classification and the fact that music encompasses various modalities such as audio, lyrics, and visuals, it is essential to exhaust all available information to learn more effectively. By integrating multiple modalities, we can capture the rich and diverse features present in music, leading to more accurate and comprehensive genre classification [Ora+18; Nan+16; Ru+23]. Furthermore, due to the scarcity of large-scale multimodal datasets in the field of MIR, it is necessary to leverage self-supervised learning to improve model performance and learn more robust representations. Self-supervised methods allow the model to generate useful training signals from the data itself, without the need for manual annotations. This is particularly valuable when dealing with multimodal data, as it enables the model to uncover meaningful relationships between modalities like audio and lyrics. By utilizing self-supervised learning, the model can learn from vast amounts of unlabeled data, leading to better generalization and more accurate music genre classification [Zha+22; Ru+23; Zhu+21; Li+24].

In this thesis on multimodal music genre classification, both audio and lyrics were utilized to capture a richer and more comprehensive representation of music. Audio provided essential acoustic features such as rhythm, timbre, and harmony, which were critical for distinguishing between genres. However, genres also carried cultural and thematic significance, often conveyed through lyrics. Lyrics contained semantic and linguistic cues that can reveal the emotional tone, narrative style, or even social context of the music. By combining these two modalities, this approach captured both the sonic and textual elements of music, offering a more holistic method for genre classification, where both aspects contributed to a deeper understanding of genre boundaries.

For the self-supervised learning component, a combination of contrastive learning and masked data modeling was employed. Contrastive learning aligned two heterogeneous modalities, namely audio and lyrics, in our multimodal approach, so as to learn the intermodal relationship. Masked data modeling further enhanced the model's ability to predict missing parts of the data, encouraging a deeper understanding of the underlying structures within both audio and lyrics. Together, these methods allowed the model to learn robust and informative representations without the need for labeled data, making them suitable for large, unlabeled music datasets.

The thesis was centered around the following research questions:

- **RQ 1** Is it complementary to utilize a multimodal learning method with audio and lyrics in music genre classification, compared to unimodal method based on audio or lyrics alone? Moreover, is audio more important than lyrics in music genre classification, or vice versa?

- **RQ 2** Does self-supervised learning enhance the performance of our model? Does contrastive learning and masked data modeling perform well when applied respectively, and is it more helpful to combine them together?
- **RQ 3** There are various hyperparameters in the proposed method, such as the input length of audio and lyrics, size of model and datasets, batch size in contrastive learning as well as masking ratio in masked data modeling. How do they affect the model performance?

The contributions of our research were as follows:

- We developed a multimodal, self-supervised model framework using audio and lyrics, which could be utilized in various downstream tasks including but not limited to music genre classification. We found out that it was complementary to combine two modalities of audio and lyrics, as well as two self-supervised learning methods – contrastive learning and masked data modeling in music genre classification. Multimodal self-supervised learning is still a novel approach in MIR. To the best of our knowledge, we were the first that combined contrastive learning and masked data modeling, and the first to apply a masked auto-encoder (MAE) [He+22] architecture in the music genre classification problem.
- We explored different choices of relevant hyperparameters, and acquired some interesting findings. For example, we found out that audio and lyrics had discrepant importances for different genres. We also discovered that the optimal masking ratio for audio and lyrics was lower than the settings in other research of masked audio/language modeling. Furthermore, masking ratio also had an impact on the performance of contrastive learning.
- Lastly, we also created a multimodal music audio-lyrics dataset containing 113,589 samples based on the Million Song dataset [Ber+11] and the Genius Song Lyrics dataset [Nay22].

Chapter 2

Background

This chapter provides preliminaries to the two important topics of this thesis: self-supervised and multimodal learning, as well as a brief introduction of the basis of our model: transformer.

2.1 Transformers

In our model, all the encoders and decoders were based on the encoder part of the transformer model [Ash+17]. Therefore, we first introduce the architecture of transformer (encoder) as an important preliminary.

The main component of the transformer model is the multi-head attention module consisting of several scaled dot-product attention layers running in parallel. The attention function in essence is mapping a series of queries and key-value pairs to an output. To compute the queries Q , keys K and values V , we transform the input $X = (x_1^T, \dots, x_n^T)^T \in \mathbb{R}^{n \times d}$ with linear transformation:

$$Q = XW_Q \in \mathbb{R}^{n \times d_k}, \quad (2.1)$$

$$K = XW_K \in \mathbb{R}^{n \times d_k}, \quad (2.2)$$

$$V = XW_V \in \mathbb{R}^{n \times d_v}, \quad (2.3)$$

where $W_Q \in \mathbb{R}^{d \times d_k}$, $W_K \in \mathbb{R}^{d \times d_k}$ and $W_V \in \mathbb{R}^{d \times d_v}$ are weight matrices. The output of the scaled dot-product attention is a weighted sum of the values V , where the weights are obtained by computing the dot product of the queries Q and keys K , divided by $\sqrt{d_k}$ and processed by a softmax function:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \in \mathbb{R}^{n \times d_v}. \quad (2.4)$$

The multi-head attention is simply a concatenation of several scaled dot-product attention layers with a linear transformation:

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O \in \mathbb{R}^{n \times d}, \quad (2.5)$$

where $W_O \in \mathbb{R}^{hd_v \times d}$ is the weight matrix, and h is the number of heads.

Each transformer encoder contains a multi-head attention $\text{Multihead}(\cdot)$ and a two-layer feed forward network $\text{MLP}(\cdot)$ sequentially. After each module, a layer normalization $\text{LayerNorm}(\cdot)$ with residual connection is applied.

$$X' = \text{LayerNorm}(\text{Multihead}(X) + X), \quad (2.6)$$

$$X'' = \text{LayerNorm}(\text{MLP}(X') + X'). \quad (2.7)$$

Since the transformer model has no convolution or recurrence, we have to manually inject the positional information into the embeddings. The positional encodings $\text{PE}(\cdot, \cdot)$ are of the same dimension d of the input embeddings, so as to be summed up together.

$$\text{PE}(\text{pos}, 2i) = \sin \frac{\text{pos}}{10000^{2i/d}}, \quad (2.8)$$

$$\text{PE}(\text{pos}, 2i+1) = \cos \frac{\text{pos}}{10000^{2i/d}}. \quad (2.9)$$

Here pos is the sequential position, and $2i$ and $2i + 1$ refer to the dimension. Using functions of $\sin(\cdot)$ and $\cos(\cdot)$, they are the so-called sinusoidal positional encodings. This definition allows the model to learn to attend by relative positions, because for any offset k , the new encoding $PE(pos + k, \cdot)$ can always be expressed as a weighted sum of two other positional encodings.

2.2 Self-supervised Learning

Deep learning is a subset of machine learning that uses neural networks with multiple layers to model complex data patterns and make predictions [LBH15]. It has powered advancements across fields such as computer vision, natural language processing, and audio processing. In supervised learning, one of the most common paradigms in deep learning, models are trained on large datasets labeled with correct outputs. By learning from these labeled examples, the model builds associations between input features and target outputs, enabling it to generalize to unseen data. However, supervised learning's dependency on labeled data can be a significant limitation, as obtaining high-quality annotations is often costly and labor-intensive.

To overcome the scarcity of labeled data, several alternative learning paradigms have been proposed, including active learning, semi-supervised learning, and self-supervised learning. Self-supervised learning, in particular, allows models to extract meaningful features from unlabeled data by solving pretext tasks, which are designed to uncover underlying structure in the data. More specifically, these pretext tasks aim to recover parts of the original input which have been corrupted, distorted or transformed [Liu+21b]. This paradigm not only reduces the need for labeled data but also enables the creation of robust and transferable features for downstream tasks. Self-supervised learning unveils inherent structure directly from unlabeled data by solving a pre-defined pretext task. By leveraging extensive unlabeled data, self-supervised learning not only eases the burden of human annotations, which are arduous and costly, but also produces robust and generalizable features for downstream tasks even from a single image [ARV20].

In the following subsections, we are going to introduce the categorization of pretext tasks, and the application of self-supervised learning in MIR.

2.2.1 Pretext tasks

Pretext tasks, also referred to as proxy tasks, are the essence of self-supervised learning. The term "pretext" indicates that solving the predefined task is not the main goal, but rather a method to produce a robust pretrained model. After reviewing several survey articles, we categorize the pretext tasks into four categories: context-based, contrastive, generative and contrastive-generative.

Context-based

Context-based tasks are mostly used in computer vision. They focus on the contextual semantics of image pixels, such as spatial structures, by predicting the geometric transformation performed on the image data. Some notable examples include rotation, colorization and jigsaw. The rotation operation rotates the image randomly by either 0° , 90° , 180° or 270° , and a convolutional neural network (CNN) model is trained to predict the angle as a four-class classification problem [GSK18]. In the colorization task, images are transformed into grayscale, and the lightness of each pixel is given to the model to predict the color [LMS17]. Lastly, there is jigsaw puzzle, which discretizes the image into patches and shuffle them, and the task itself is to recognize the original order [NF16]. The complexity of this task can be further increased by combining it with rotation [Li+20].

Contrastive

The aim of contrastive learning is (a) to bring semantically similar samples closer together in the latent space while (b) maximizing the distance between dissimilar ones. In this regard, a number of self-supervised contrastive learning methods are developed. The most representative among them is built upon negative samples, exemplified by MoCo (Momentum Contrast) (v1 [He+20]),

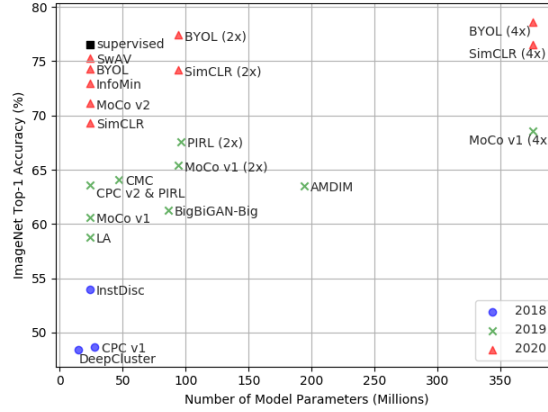


Figure 2.1: The performance of self-supervised learning methods on ImageNet top-1 accuracy (i.e, the conventional classification accuracy in which the prediction with the highest accuracy must be the exact ground truth.) in March 2021 [Liu+21b]. The horizontal axis was the model size, in terms of the number of parameters; the vertical axis was the accuracy, which reflected the model performance. The ability of many self-supervised contrastive learning methods (e.g., SimCLR, MoCo v2, BYOL, SwAV) were almost comparable to, or even surpassed the supervised learning method (ResNet 50).

v2 [Che+20c]) and SimCLR (v1 [Che+20a], v2 [Che+20b]). These methods even achieved comparable performance to that of supervised learning with only two times of model parameters (Figure 2.1). As the term "contrastive" suggests, positive samples are brought closer and negative ones are separated as much as possible, which also forms an instance discrimination task. Although the specific definition of positive and negative samples varies depending on the scenario (e.g., multi-modal learning), the principle is that the views derived from the same instance (called the anchor sample) are treated as positive samples, and those generated from different instances are always considered as negative.

To create various diverse positive samples, data augmentation strategies are utilized. For example, SimCLR generated 10 different views of each image using cropping, resizing, flipping, rotation, color distortion, cutout, etc. Data augmentation techniques have an remarkable contribution to the performance of self-supervised contrastive learning. SimCLR and MoCo v2 both found that their models benefited from stronger data augmentation substantially, which created a more challenging and diverse learning environment with sufficient negative samples, in order to avoid finding collapsed representations.

To realize the goals (a) and (b), a contrastive loss is commonly used as objective of minimizing. Here we also take SimCLR as an example, in which they named their loss function as NT-Xent. For every positive sample pair (x_i, x_j) , a pairwise loss is defined as

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(x_i, x_j)/\tau)}{\sum_{k=1, k \neq i}^{2N} \exp(\text{sim}(x_i, x_k)/\tau)}, \quad (2.10)$$

where $\text{sim}(\cdot, \cdot)$ is cosine similarity, and τ is the hyperparameter of temperature. It controls the sharpness of the similarity distribution between positive and negative pairs. Lower values of τ lead to sharper distributions, emphasizing stronger similarities for closer pairs and reducing similarities for farther pairs, which makes the model more discriminative. Conversely, higher values of τ produce smoother distributions, making it easier for the model to assign moderate similarity to a wider range of samples. The numerator in Equation 2.10 contains the similarity of the positive sample pair, which is going to be maximized (goal (a)), while the denominator entails all negative sample pairs with respect to the anchor sample x_i , which will be minimized (goal (b)). In a batch $\{x_i\}_{i=1}^N$ of size $2N$, which is augmented from a batch $\{x_{2i-1}\}_{i=1}^N$ of size N , every positive sample

pair can be denoted as (x_{2i-1}, x_{2i}) ($i = 1, \dots, N$). The final loss is summed up as follows (note that $\mathcal{L}_{i,j}$ is not symmetric):

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^N (\mathcal{L}_{2i-1,2i} + \mathcal{L}_{2i,2i-1}). \quad (2.11)$$

Self-supervised contrastive learning has been highly effective in learning representations that capture meaningful similarities and differences in data. By focusing on learning invariant features that are robust to various augmentations, contrastive learning can improve the generalization of the model to new, unseen data, thus the learnt representations can be transferred to many downstream tasks with minimal supervision, often achieving state-of-the-art performance.

However, negative-sampling-based contrastive learning is prone to several drawbacks. Firstly, the effectiveness of contrastive learning relies heavily on selecting appropriate negative samples. If the negative samples are not sufficiently diverse or are too easy to distinguish, the model may not learn meaningful representations. Conversely, if negative samples are too similar to the positive pairs (known as false negatives), they may confuse the model, leading to suboptimal learning. Secondly, in large datasets, the number of potential negative samples can be very large. Some contrastive learning methods also rely on large batch sizes to ensure sufficient negative samples within each batch. Efficiently sampling and comparing against a vast number of negatives can be computationally expensive. Finally, identifying and leveraging hard negatives (negatives that are difficult to distinguish from positives) is important for effective training. Nevertheless, mining these hard negatives can be complex and also computationally intensive.

Therefore, some researchers attempted to discard negative sampling and attained exceptional outcomes. In other words, goal (b) no longer applies. There are mainly two kinds of methods: self-distillation-based methods, such as BYOL [Gri+20], SwAV [Car+20] and SimSiam [CH21], as well as feature decorrelation-based methods including Barlow Twins [Zbo+21] and Variance-invariance-covariance regularization (VICReg) [BPL21]. Since they are not relevant to our proposed method in this thesis, a detailed introduction will not be given.

Generative

In this section, we only discuss masked data modeling (MDM) (or referred to as masked image modeling in computer vision), as it is the most popular method in generative self-supervised learning, and most relevant to our study. MDM randomly masks some tokens in the sequence and attempts to predict the masked tokens from the corrupted input. Therefore, we define the loss function of MDM as

$$\mathcal{L}_{\text{recon}}(D(E(M(I))), I), \quad (2.12)$$

where $\mathcal{L}_{\text{recon}}$ is the reconstruction loss (e.g., mean square error), I is the input, $M(\cdot)$ is the masking operation, $E(\cdot)$ is the encoder and $D(\cdot)$ is the decoder. Notably, neural language models such as BERT [Dev+19] and its variants first achieved remarkable success in natural language processing. BERT adopted a low masking ratio, where 15% of words were randomly masked and predicted based on their context, which is similar to the cloze task [Tay53]. However, extending the success of MDM to computer vision was once challenging, mainly because of two reasons. Firstly, image signal is continuous in its nature, while MDM first needs to transform the input into discrete tokens to enable a large-scale pretraining. Secondly, the transformer architecture is widely utilized in NLP, which is scalable to both dataset and model size, while in CV, CNN was the only dominant architecture in many years. The situation evolved when vision transformer (ViT) [Dos20] emerged as the pioneer work to apply the transformer architecture and large-scale pretraining with MDM to the field of CV. ViT split the image into patches and projected them into a common latent space to create tokens. However, despite the progress it made, ViT still fell short of supervised pretraining, motivating further researches. Following the route of ViT, the MAE (Figure 2.2) revealed that a surprisingly high masking ratio (75%) was critical for images. Images often contain repetitive or highly correlated information across patches. A low masking ratio allows the model to learn short-cut patterns or rely on local details rather than developing a deeper understanding of the global structure. Similarly, SimMIM, a popular framework for masked image modeling, also directly operated on raw pixels and patchify the images into tokens. In contrast, BEiT [Bao+21] introduced an extra tokenization procedure by first training a discrete variational autoencoder (dVAE) to create a

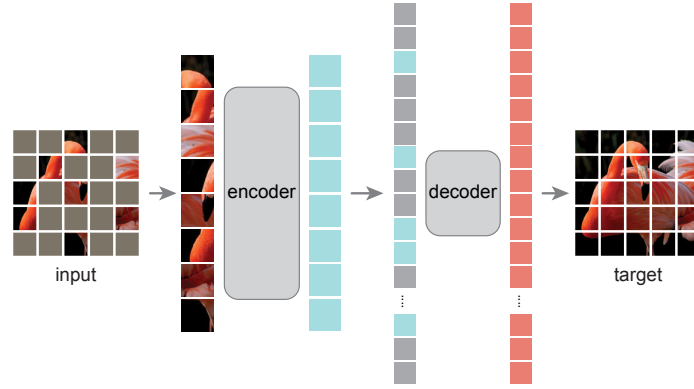


Figure 2.2: The architecture of MAE. Images are patchified and a large proportion of patches (75%) are masked then reconstructed [He+22].

pre-defined visual vocabulary. Nevertheless, this approach is more complicated compared to the end-to-end MAE, because it requires an extra model as the tokenizer.

Contrastive-generative

We have concluded in the previous sub-section that contrastive self-supervised learning is, in many cases, data-hungry and resource-intensive, whereas MDM could be inferior in its ability of data scaling. Xie et al. found that, after a certain point, increasing the size of the dataset may not lead to proportional gains in the performance of MDM, especially under a non-overfitting scenario [Xie+23]. What is more, MDM models mostly low-level semantics because of its point-wise nature ("point" may refer to pixel in images, word in text, etc.), while many classification tasks target at high-level global information, which is exactly the focus of contrastive learning [Liu+21b]. Therefore, it may be complementary to combine contrastive and generative (MDM) self-supervised learning, which motivates the researchers to make several endeavors, including iBOT [Zho+21], RePre [Wan+22a], CMAE [Hua+23b], CAV-MAE [Gon+23] and SiameseIM [Tao+23]. For instance, the CMAE (Contrastive Masked Autoencoder) model used two branches – the online branch reconstructed masked image patches, and the momentum branch operated on the full image and generated embeddings to guide the online branch through a contrastive learning objective. This approach enhanced the model's ability to generalize by jointly optimizing for both local image reconstruction and global semantic consistency. Quite similarly, SiameseIM used a Siamese network architecture with two identical branches to accept different augmented views of an image. The online branch encoded the first view which was masked, and predicted the second view based on the first view. The target branch encoded the second view and compared it with the prediction. Both two methods witnessed improvements on downstream tasks, compared to baselines with contrastive learning or masked image modeling alone, such as MoCo, BEiT and MAE.

2.3 Multimodal Learning

In machine learning, modality refers to a distinct source or type of information that represents different ways of perceiving or interpreting data. Multimodal learning is a field of machine learning that involves integrating and processing information from multiple data modalities, such as text, images, and audio. By leveraging the potentially complementary strengths of different modalities, multimodal learning models aim to improve performance on tasks where single-modality approaches may fall short. The challenge lies in effectively fusing and aligning information from these diverse sources to capture richer and more holistic representations.

2.3.1 Modality fusion and alignment

In multimodal learning, each modality captures different aspects of the same underlying information. Modality fusion refers to the process of integrating data from different modalities to create a unified representation that can be used by machine learning models. There are mainly two basic architectures of modality fusion in the pretraining stage: late fusion with modality-specific encoders and early fusion with a unified encoder [ZMH23]. Furthermore, there is also a so-called hybrid or intermediate fusion, which combines elements of both early and late fusion.

Late fusion involves processing each modality independently with modality-specific encoders and then combining the results at the decision level. Formally, we define $x = (x_1, \dots, x_k)$ as a data point consisting of k different modalities, $e_i(\cdot)$ ($i = 1, \dots, k$) as corresponding encoders, $f(\cdot)$ as the modality fusion module and $g(\cdot)$ as the classification head, then a model $h(x)$ with late modality fusion can be written as

$$h(x) = g(f(e_1(x_1), \dots, e_k(x_k))). \quad (2.13)$$

Recent late fusion methods often enforce a similarity measure like contrastive loss, or employ a multimodal transformer as the fusion module to align the modalities. We will discuss this later.

Early fusion methods, in contrast, use a unified encoder to process the inputs of all heterogeneous modalities:

$$h(x) = g(f(x_1, \dots, x_k)). \quad (2.14)$$

Many contemporary early fusion methods also leverage a transformer-based fusion module, which still requires modality-specific tokenization, such as patching embeddings for visual input, and word embeddings for text input. A unified encoder can implicitly learn intermodal relationships and intramodal information at the same time, which will be proved at the end of Chapter 4. To prevent a confusion between the inputs of different modalities, we may use positional encodings and modality type encodings along with the inputs. Moreover, the design of unified encoder has higher flexibility, allowing the missing of some modalities.

Apart from modality fusion, another challenge is to map or pair the data from different modalities correctly, so that they correspond to the same underlying concept or instance, which is referred to as modality alignment. It can be categorized into two kinds, namely coarse-grained alignment which pairs the data on the level of instances, as well as fine-grained alignment which links components of tokens or instances [ZMH23] (Figure 2.3).

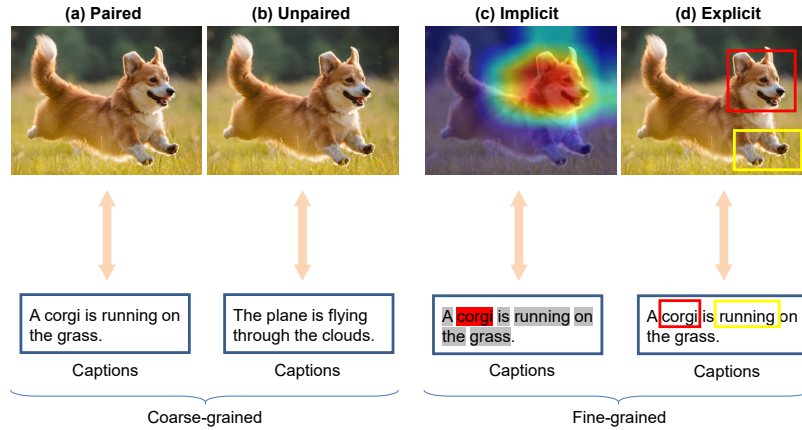


Figure 2.3: The categorization of modality alignment [ZMH23].

For coarse-grained alignment, contrastive learning is often used as an instance discrimination (ID), where different modalities from corresponding samples are considered as positive samples and those from different samples are considered as negative ones. When the temporal dimension is considered in video data, different frames from the same video can even be considered as negative, too. However, the pairing of data is noisy in many cases. For example, in image-text dataset, some objects in an image may not be in the text, and an event described in the text may not be perfectly visible in the image. Such noisyness negatively impacts the performance of contrastive learning.

When it comes to fine-grained alignment, one can also further divide it into explicit and implicit methods. The former is between corresponding components within each modality, such as an object in the image to a word in the caption text, while the latter learns a soft association across tokens in different modalities. In this section, we only cover the implicit alignment as the explicit alignment is irrelevant to the proposed method. Cross-attention (or co-attention), an extension of self-attention (see Equation 2.4) in multimodal scenario, is the most commonly used approach in implicit alignment, because its mechanism is consistent with the idea of fine-grained implicit alignment. To jointly attend two modalities, we only need to exchange their key-value pairs:

$$\text{Attention}(Q_1, K_2, V_2) = \text{softmax} \left(\frac{Q_1 K_2^T}{\sqrt{d}} \right) V_2, \quad (2.15)$$

$$\text{Attention}(Q_2, K_1, V_1) = \text{softmax} \left(\frac{Q_2 K_1^T}{\sqrt{d}} \right) V_1, \quad (2.16)$$

in which (Q_1, K_1, V_1) and (Q_2, K_2, V_2) are the query, key and value matrices of modality 1 and 2, respectively. d refers to their latent dimension. In this way, the model can learn the cross-modal interactions between each pair of tokens from different modalities by calculating the dot products $Q_1 K_2^T$ and $Q_2 K_1^T$. The cross-attention was first proposed in [Ren+16], and has been widely applied in multimodal learning.

2.3.2 Multimodal self-supervised learning

Self-supervised multimodal learning is essential because it leverages the vast amounts of unlabeled data across multiple modalities to learn rich and generalized representations without the need for extensive manual annotation. By exploiting naturally occurring correlations between modalities, self-supervised learning allows models to learn useful features from one modality to predict or understand another, thus enhancing performance on a wide range of tasks. Deriving from self-supervised learning under unimodal scenarios, multimodal self-supervised learning adapt the original objective functions to take different modalities into account. Thus, it can be also classified as contrastive, generative (i.e., MDM in this thesis) and contrastive-generative [ZMH23].

Contrastive

Multimodal contrastive learning aligns different modalities by bringing the paired modalities from the same instance closer and pushing the unpaired ones further apart (Figure 2.4). Given a multimodal dataset with two modalities: $\left\{ \left(x_1^{(i)}, x_2^{(i)} \right) \right\}_{i=1}^n$, where n is the number of samples, then the contrastive loss of a data point $\left(x_1^{(i)}, x_2^{(i)} \right)$ is

$$\mathcal{L}_i = -\log \frac{\exp \left(\text{sim} \left(x_1^{(i)}, x_2^{(i)} \right) / \tau \right)}{\exp \left(\text{sim} \left(x_1^{(i)}, x_2^{(i)} \right) / \tau \right) + \sum_{k=1, k \neq i}^n \exp \left(\text{sim} \left(x_1^{(i)}, x_2^{(k)} \right) / \tau \right)}. \quad (2.17)$$

CLIP [Rad+21] was one of the most classic and successful works in image-language joint pre-training, exemplifying the great scalability of multimodal contrastive learning. It was pretrained on a huge dataset with 400M samples, and demonstrated strong zero-shot (predicting samples with classes never observed during training) performance on transferring to various downstream task. Other examples include AVTS [KTT18], ASTA [MLN20] (audio-video), Multimodal Versatile Networks [Ala+20] and Video-Audio-Text transformer (VATT) [Akb+21] (audio-video-text).

Generative

MDM in a multimodal context often predicts the masked information of one modality by conditioning on another modality. In this way, the model is forced to encode the relationship be-

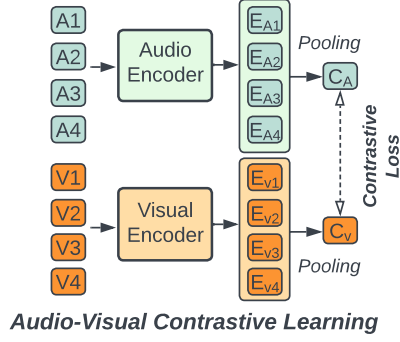


Figure 2.4: An example of multimodal contrastive learning from the paper of CAV-MAE [Gon+23].

tween different modalities (Figure 2.5). Given a dataset $\left\{ \left(x_1^{(i)}, x_2^{(i)} \right) \right\}_{i=1}^n$ and its masked version $\left\{ \left(\tilde{x}_1^{(i)}, \tilde{x}_2^{(i)} \right) \right\}_{i=1}^n$, a reconstruction loss of modality 1 given modality 2 is formed as

$$\mathcal{L}_1 = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_{\text{recon}} \left(D \left(E \left(\tilde{x}_1^{(i)}, \tilde{x}_2^{(i)} \right) \right), x_1^{(i)} \right). \quad (2.18)$$

In addition, MDM is more flexible to handle the missing of modalities compared to contrastive learning, and can be performed under a unimodal setting. Notable examples of multimodal MDM include VideoBERT [Sun+19], VL-BEiT [Bao+22a], BEiT-3 [Wan+22b] and M3AE [Liu+23].

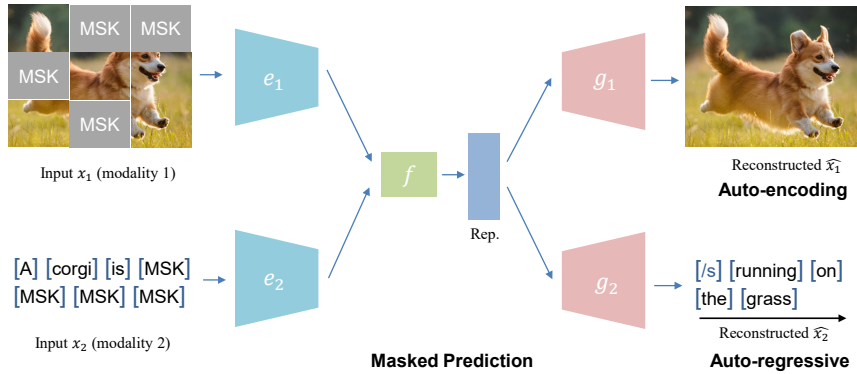


Figure 2.5: An illustration of the multimodal MDM, where the model predicts the corrupted input of one modality based on the information of the other [ZMH23].

Contrastive-generative

The combination of contrastive learning and MDM in multimodal learning is an emerging approach, but it has also recently attracted the attention of researchers. Generally, it can be viewed as a multi-task learning problem, in which the objective function is a weighted sum of two separate objectives:

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_m, \quad (2.19)$$

where \mathcal{L}_c and \mathcal{L}_m are the loss function of contrastive learning and MDM, respectively, and λ refers to the weight. By jointly optimizing both the objectives, we are able to leverage the complementary power of two different self-supervised learning methods in a multimodal scenario. However, it is also possible that different objectives may be conflict in the optimization process, where optimizing one of them is against the other.

Examples have emerged in various fields, such as ALBEF [Li+21], VLMO [Bao+22b] as well as FLAVA [Sin+22] for image-text pretraining, MERLOT Reserve [Zel+22] for video-text pretraining, and CAV-MAE, MAViL [Hua+23a] for video-audio pretraining. For instance, ALBEF employed

contrastive learning before fusing image and text with cross-attention, and predicted the masked text with complete image tokens. Similarl, FLAVA also optimized the combination of contrastive and MDM loss, while being able to handle unpaired modality with intramodal MDM. CAV-MAE masked both image (video) and audio, and computed contrastive loss between two masked modalities. Afterwards, the unmasked tokens of image and audio were concatenated to predict the masked ones. Moreover, an interesting observation is that most of these models adopted a similar architecture with attention-based, modality-specific encoders followed by a unified modality fusion encoder.

Chapter 3

Related Work

In this chapter, we present a few research work related to the idea of this thesis. We first discuss the application of self-supervised learning in the field of MIR, then we introduce some literatures classifying music genres using either music audio or lyrics, as well as those utilizing a multimodal approach with both two modalities.

3.1 Self-supervised learning in MIR

The field of MIR has long faced challenges with the scarcity of dataset, primarily due to the high costs of music audio annotation and the restrictions imposed by country-specific copyright laws. Self-supervised learning holds great potential for addressing this issue. By leveraging large amounts of unlabeled music data, self-supervised learning can automatically extract useful features without the need for costly manual annotation. However, the application of self-supervised learning in the field of MIR remains relatively limited.

Previously, there were a few context-based self-supervised learning methods applied to music. Wu et al [Wu+21]. proposed a multi-task learning approach to predict multiple hand-crafted audio features, such as log power spectrum (LPS), Mel-frequency cepstral coefficients (MFCC), Chroma, and Tempogram. They found that a proper combination of tasks with curated weights resulted in higher performance of downstream tasks like instrument and genre classification. Carr et al. [Car+21] designed a jigsaw task in which a spectrogram was split into patches and then shuffled. The model was pretrained by predicting the permutation. The experimental results unveiled an interesting finding that splitting along the frequency dimension led to the strongest improvement of performance. Recently, researchers discovered new possibilities of context-based methods in MIR tasks, such as learning equivariance of music features – which means an transformation on the input is equivalently reflected on the model output. By enforcing an equivariance, the model can be sensitive to the changes of desired features. Two examples were [Qui22] for tempo estimation, [Kon+24] for tonality estimation and [Rio+23] for pitch estimation. They both adopted a Siamese network architecture and carefully designed loss functions to capture the equivariance. Surprisingly, the latter achieved comparable results to supervised baselines with a light-weight model with only 1/1000 parameters. Zero-Note Samba [DLH23] also used a Siamese network and an external source separation model to predict the similarity of beat tracking between the percussive and non-percussive parts of musical pieces. If two parts were synchronized, then the similarity should be high, or vice versa.

Contrastive learning was another dominant self-supervised learning method in MIR. Many earlier works were SimCLR-like, based on negative sampling. Various data augmentation techniques were employed to create different views for the model. For instance, CLMR proposed by Spijkervet et al. [SB21] extended SimCLR to music domain, performing a series of data augmentations such as polarity inversion, Gaussian addition, frequency filter on raw audio waveforms. The results were significant: the performance of the tag prediction task was comparable or even surpassed the supervised baselines. Focusing on singing voices, Yakula et al. [YWG22] also utilized data augmentation (time stretching and pitch shifting) but in a reverse way: the network was trained to push the transformed version away. In this way, the model became attentive to vocal timbre and singing

expression, which is useful in the singer identification task. Garoufis et al. [GZM23] modified the contrastive audio representation learning framework COLA to utilize an external source separation model and consider sources extracted from the same piece as positive samples. It turned out that their model outperformed the original COLA which produced different views piece-wise in the downstream music tagging task. Recently, Torres et al. [TLR23] took a pioneering step by exploring novel contrastive learning frameworks that were not based on negative sampling, such as BYOL and VICReg, in the singer identification task. Results showed that BYOL behaved especially well in some out-of-domain datasets while operating at 44.1 kHz. Lastly, some researchers innovatively proposed a method where contrastive learning could also be used to evaluate the coherence of tracks in music accompaniment generation by calculating the output similarity score [Cir+24].

MDM has rarely been used in MIR so far [Zhu+21; Li+24]. However, a model called MERT [Li+24] claimed that it attained state-of-the-art in the average score of 14 various downstream tasks, especially those focused on local information such as pitch, rhythm and timbre (singer identification). MERT utilized two teacher models, namely one acoustic and one musical. The acoustic model was derived from HuBERT [Hsu+21], in which the authors compared two settings: one was offline clustering of the log-Mel spectrum and Chroma features to get the pseudo-labels. The pseudo-labels were then predicted given the masked input. The other used residual Vector Quantized-Variational AutoEncoder (VQ-VAE). The musical model aimed to reconstruct the Constant-Q Transform (CQT) spectrogram, in order to capture the pitch information.

3.2 Classifying music genres with audio and text

3.2.1 Music genre classification with audio

From the first day when the music genre classification problem was raised, audio has always been the most important and widely-used modality. Hand-crafted audio-based features can be roughly divided into three categories, namely timbre, pitch and rhythm.

A large group of timbre features are based on the statistics of results of short time Fourier transform (STFT) performed on the audio signal, such as Zero Crossing Rate (ZCR) [TC02; LOL03; Ber+06], Spectral Centroid (SC) [TC02; LOL03; Ber+06], Spectral Rolloff (SR) [TC02; LOL03; Ber+06], Spectral Flux (SF) [TC02; LOL03], and Spectral Bandwidth (SB) [TC02; Ber+06]. To better simulate human's perception of audio signal, which has finer resolution in lower frequency range, the spectrogram is usually decomposed into subbands and scaled logarithmically. Such a strategy has achieved success in music-related tasks including music genre classification, with typical examples like Mel-frequency Cepstrum Coefficient (MFCC) [TC02; PN17; LC11; LOL03], Amplitude Spectrum Envelope (ASE) [Lee+09], Octave-based Spectral Contrast (OSC) [Lee+09] and Daubechies Wavelet Coef Histogram (DWCH) [Wan+09; LOL03].

Pitch is another important auditory feature. It is mainly determined by the fundamental frequency of a note but can be affected by multiple factors like loudness, timbre and its harmonic series, as pointed out in [Fu+10]. Therefore, pitch detection, especially multi-pitch detection is a challenging task. To mitigate this problem, pitch histogram has been applied to estimate the statistical information of pitches on the level of a song [TC02; TEC03]. Besides, pitch class profile, or the chroma feature which describes the relative position of a note within an octave, is also used along with other timbre features such as MFCC in music classification tasks [Eli07; SLT19; SS15].

Rhythm describes a repeated pattern of movement and silence in music. Commonly used rhythm features entail two properties, namely beat and tempo (beat per minute) [Fu+10]. Similarly to the pitch features, beat histogram detects the tempo and calculates relevant statistics to model a periodicity distribution representation. Some examples of its application in music genre classification include [TC02; BGL15; LL20]. Nevertheless, this approach has a main drawback that rhythm is prone to be affected by different sound properties, such as amplitude, spectral and tonal changes [LL20]. Therefore, [LL20] tackled this shortcoming by extracting novelty functions from these properties to take their influence into account.

Before the era of deep learning, these hand-crafted features were no doubt the mainstream of MIR, along with traditional machine learning models. The list included but was not limited to linear regression [AKR15; Bah18], linear discriminant analysis [LOL03], gaussian mixture model

[LOL03; TC02], naïve Bayes [SR15; KRR18; ZMM17; AKR15], k-nearest-neighbour [TC02; TEC03; WM21; AKR15], as well as some more advanced algorithms, such as support vector machine [KRR18; SR15; AKR15; Elb+18; Ful+18; CJ13; Nan+16], ensemble learning (random forest) [AKR15; KRR18; SR15; Elb+18] and boosting algorithms [KRR18] (XGBoost) [CJ13; Nan+16] (AdaBoost).

However, hand-crafted features require an expertise in the field of music and sound, which hinders the participation of researchers from various backgrounds. Meanwhile, recent years have witnessed the rise of deep learning which proposes an end-to-end framework without the need of hand-selecting features and classifier, and achieves tremendous success in computer vision, natural language processing and other fields. This results in a shift of focus from feature engineering to model design and data collection. Under this circumstance, spectrograms come in useful for the direct application of deep learning methods, especially computer vision models, due to their similarities to images. STFT and Mel spectrograms have become mainstream of recent works with a deep learning framework [SD14; Zha+16; Cho+17a; Cho+17b; COS17; Meh+21].

Just as in the field of computer vision, convolutional neural networks (CNNs) are largely applied to spectrogram data. For instance, [Meh+21] made a comparison between some classic CNN architectures: ResNet, VGG and AlexNet, and found that the best-performing ResNet34 surpassed the support vector machine on GTZAN dataset. Recently, some novel network architectures were also utilized for the music genre classification task. [Liu+21a] designed an architecture called broadcast module, featuring skip connections between inception blocks consisting of convolutions with different kernel sizes, in order to take the long contextual information into consideration. This proposed network outperformed several CNNs with more traditional architectures on three datasets: GTZAN, Ballroom and Extended Ballroom. Apart from CNN, recurrent neural network (RNN), including long short-term memory (LSTM) and convolutional recurrent neural network (CRNN), was another popular model of the music genre classification task [WM21; Cho+17a; Elb+18; Ful+18; Yan+20]. Moreover, since transformer-based models have achieved remarkable success in many fields, some researchers also attempted to incorporate attention mechanism into their models [Yu+20; Gan21; Zha21; ZCZ20]. For example, [Yu+20] proposed a serial attention model based on bidirectional recurrent neural network (BRNN) and a parallelized attention model based on CNN, and found the latter was superior to the former on STFT spectrograms.

3.2.2 Music genre classification with lyrics

Lyrics, containing rich semantic information, are also useful in distinguishing songs of similar music styles [MR10], and there are a number of works focusing on lyrics-based music genre classification. Similar to audio-based features, bag of words (BoW) and n-gram were the most common features to acquire statistical information from lyrics before the success of deep learning models, exemplified by [ÇÖ16; MNR08b]. Besides, linguistic features, such as rhyme features and Part-Of-Speech (POS) were also used in early works [MNR08b; FS14]. However, these techniques only work at the lexical level and have limited capacity in capturing the semantics. The development of deep learning models and neural word embeddings facilitates the learning of high-level semantic structures in textual data [KRR18; Tsa17; Ara+20]. [Tsa17] adapted a Hierarchical Attention Network (HAN) operating on the GloVe [PSM14] embeddings to discover the hierarchical structures in lyrics on different levels of words and sentences. The experimental results showed that the proposed model outperformed both neural and non-neural models, such as linear regression and LSTM. As an example of non-English lyrics classification, [Ara+20] classified Brazilian song lyrics using pretrained Portuguese word embeddings [Har+17] based on Word2Vec [Mik+13]. The paper utilized a Bidirectional Long Short-Term Memory (LSTM) which captured both past and future information in the context, and found it significantly outperformed two machine learning models, SVM and random forest.

3.2.3 Multimodal music genre classification with audio and lyrics

To the best of our knowledge, [MNR08a] was the first to combine audio and lyrics in music genre classification. The authors experimented on a variety of choices on possible features of both modalities, such as Statistical Spectrum Descriptors (SSD) and rhythm features for audio, and those in

[MNR08b] for lyrics. The results showed that a multimodal combination of SSD and text statistics features significantly improve the classification accuracy compared to the SSD-alone baseline. Later, the authors also enhanced the fusion method by applying a late fusion with a Cartesian Ensemble System [MR11]. Similarly, [VM22] conducted an in-depth exploration of the individual impact of six different feature groups (including audio and lyrics) on different categories, and applied a multi-objective feature selection strategy using evolutionary algorithms.

However, the complexity of multimodal learning and the diversity of modalities make the use of more advanced models highly necessary. For instance, [Pan+21] proposed a model with a residual CNN as the audio network and a HAN as the lyrics network, following by a feature concatenation as the late fusion of modalities. [Li+23] also used CNN for audio spectrogram but used a more sophisticated BERT for lyrics. Nevertheless, the modality fusion mechanisms of these works were rather naive – either a direct concatenation, or a weighted sum of the representations, which were prone to some problems like modality imbalance and lack of interaction modeling. [WM21] improved this deficiency by utilizing multiple dense co-attention layers as the modality fusion, and outperformed a weighted sum fusion by more than 5% in accuracy. Recently, [Ru+23] made a remarkable progress in the multimodal music genre classification task. They innovatively introduced self-supervised contrastive learning to align the heterogeneous modalities of audio and lyrics, and leveraged a cross-modal attention for modality fusion, and found both of them beneficial to the model performance.

Chapter 4

Methods

In this chapter, we are going to introduce our model in detail. Inspired by CAV-MAE [Gon+23], we proposed a network architecture that enables the multimodal learning of music audio and lyrics, as well as a combination of two self-supervised learning methods: contrastive learning and masked data modeling.

4.1 Tokenizing audio and lyrics

To enable the use of transformer-based encoders on audio and lyrics data, we had to convert the data into sequences of tokens. For music audio, we followed the tokenization method in audio spectrogram transformer (AST) [GCG21]. For an audio spectrogram, we split it into n non-overlapping square patches of size $s \times s$ as the input of the model: $\bar{\mathbf{a}} = [\bar{a}_1, \dots, \bar{a}_n]$ (Figure 4.1). The lyrics were tokenized into words using a tokenizer (e.g., BERT tokenizer), then padded or truncated to a uniform length m . To facilitate a joint encoder, the audio patch sequence $\bar{\mathbf{a}}$ was projected into a latent space of dimension d :

$$\mathbf{a} = \text{Proj}_a(\bar{\mathbf{a}}) \in \mathbb{R}^{d \times n}, \quad i = 1, \dots, n, \quad (4.1)$$

which was of the same dimension of the lyrics embeddings: $\mathbf{t} = [t_1, \dots, t_m] \in \mathbb{R}^{d \times m}$.

4.2 Self-supervised pretraining

As a two-stage model, the self-supervised learning methods were utilized in the pretraining stage. The whole model for pretraining mainly comprised two modal-specific encoders: E_a and E_t , a joint encoder E_j and a joint decoder D_j . For an illustration of the model in the pretraining stage, please check Figure 4.2.

4.2.1 Audio-lyrics MAE

We applied masked data modeling in the fashion of multimodal MAEs. The preprocessed audio and lyrics tokens \mathbf{a} and \mathbf{t} were added with a modality type embedding \mathbf{E}^M and a 2D sinusoidal

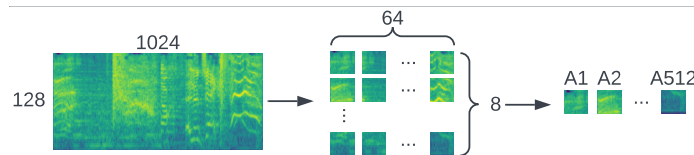


Figure 4.1: The tokenization of audio spectrograms by splitting them into non-overlapping square patches.

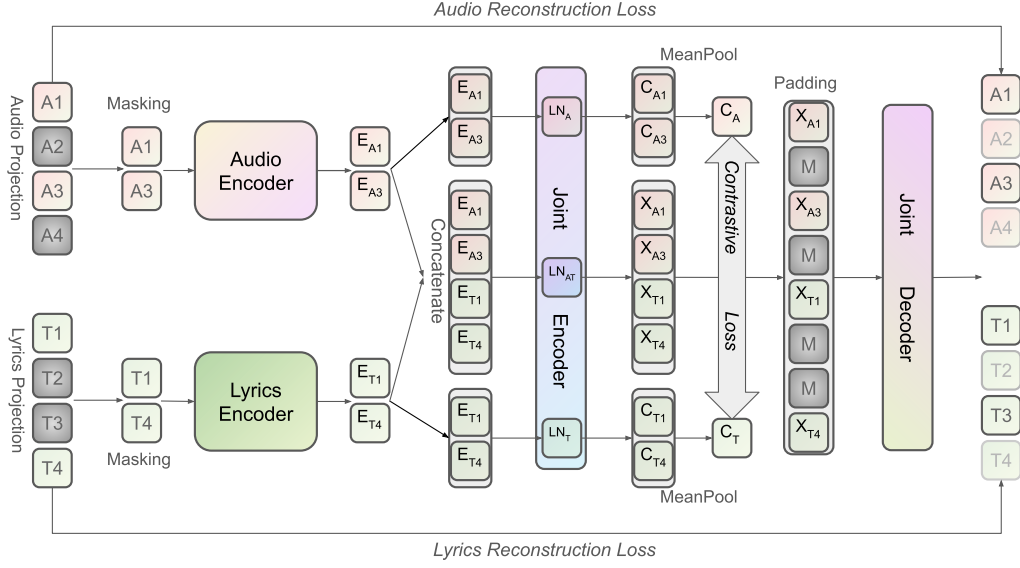


Figure 4.2: The model framework in the pretraining stage. The input sequences of audio and lyrics were masked and pass through the modal-specific encoders, and took a multi-stream data pass to the joint encoder to compute the contrastive loss and predict the masked tokens through a joint decoder.

position encoding E^P (one dimension for row, the other for column). Then, the tokens were randomly and uniformly masked with modal-specific masks M_a and M_t , which had masking ratios of m_a and m_t .

$$\mathbf{a}^{\text{unmask}} = M_a \left(\mathbf{a} + \mathbf{E}_a^M + \mathbf{E}_a^P \right) \in \mathbb{R}^{d \times (n-n')}, \quad (4.2)$$

$$\mathbf{t}^{\text{unmask}} = M_t \left(\mathbf{t} + \mathbf{E}_t^M + \mathbf{E}_t^P \right) \in \mathbb{R}^{d \times (m-m')}, \quad (4.3)$$

where n' and m' were the number of masked tokens for audio and lyrics, respectively.

The unmasked tokens were fed to the modal-specific encoders E_a and E_t , respectively, which consisted of multiple layers of transformer encoders.

$$\tilde{\mathbf{a}} = E_a(\mathbf{a}^{\text{unmask}}), \quad (4.4)$$

$$\tilde{\mathbf{t}} = E_t(\mathbf{t}^{\text{unmask}}). \quad (4.5)$$

To combine contrastive learning and masked data modeling, a multi-stream pass of data was designed between the modal-specific encoders and the joint encoder. The joint encoder E_j took the output of two modal-specific encoders $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{t}}$, as well as their concatenation $[\tilde{\mathbf{a}}, \tilde{\mathbf{t}}]$, while sharing all the parameters for all three passes, except those of the layer normalization LN. Our reconstruction task used the concatenation.

$$\mathbf{x} = E_j([\tilde{\mathbf{a}}, \tilde{\mathbf{t}}]; \text{LN}_{at}) \in \mathbb{R}^{d \times (n-n'+m-m')}. \quad (4.6)$$

The joint representation \mathbf{x} was projected into a latent space of dimension d' , padded with trainable masked tokens $\mathbf{m} \in \mathbb{R}^{d' \times (m'+n')}$ in the original masked positions, and added with modality-type embeddings $\hat{\mathbf{E}}^M$ and 2D sinusoidal position encodings $\hat{\mathbf{E}}^P$, before it was fed into the joint decoder D_j to recover the masked tokens. The joint decoder had two linear projection heads for the two modalities: $\text{MLP}_a : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$ and $\text{MLP}_t : \mathbb{R}^{d'} \rightarrow \mathbb{R}^V$ (V was the vocabulary size (number of classes) of the lyrics tokenizer).

$$[\mathbf{x}'_a, \mathbf{x}'_t] = \text{Pad}(\text{Proj}_d(\mathbf{x}), \mathbf{m}) \in \mathbb{R}^{d' \times (n+m)}, \quad (4.7)$$

$$\hat{\mathbf{a}} = D_j \left(\mathbf{x}'_a + \hat{\mathbf{E}}_a^M + \hat{\mathbf{E}}_a^P; \text{MLP}_a \right) \in \mathbb{R}^{d \times n}, \quad (4.8)$$

$$\hat{\mathbf{t}} = D_j \left(\mathbf{x}'_t + \hat{\mathbf{E}}_t^M + \hat{\mathbf{E}}_t^P; \text{MLP}_t \right) \in \mathbb{R}^{V \times m}. \quad (4.9)$$

Finally, the reconstruction loss was built upon the predictions of the masked tokens $\hat{\mathbf{a}}_k^{\text{mask}}$ and $\hat{\mathbf{t}}_k^{\text{mask}}$ ($k = 1, \dots, N$, where N was the size of the mini-batch). For audio spectrograms, we used the mean square error (MSE) loss between the prediction $\hat{\mathbf{a}}_k^{\text{mask}} \in \mathbb{R}^{d \times n'}$ and ground truth $\mathbf{a}_k^{\text{mask}} = \mathbf{a}_k \setminus \mathbf{a}_k^{\text{unmask}} \in \mathbb{R}^{d \times n'}$. The audio reconstruction loss \mathcal{L}_{r-a} was defined as follows:

$$\mathcal{L}_{r-a} = \frac{1}{N} \sum_{k=1}^N \|\hat{\mathbf{a}}_k^{\text{mask}} - \mathbf{a}_k^{\text{mask}}\|_F^2, \quad (4.10)$$

where $\|\cdot\|_F$ denoted the Frobenius norm.

The lyrics reconstruction was treated as a multi-class classification task, predicting the masked words. Naturally, we chose the cross-entropy loss function. Denote

$$\hat{\mathbf{t}}_k^{\text{mask}} = [\hat{t}_k^{(1)}, \dots, \hat{t}_k^{(m')}] \in \mathbb{R}^{V \times m'}, k = 1, \dots, N \quad (4.11)$$

as the predicted probabilities for the k -th sequence of masked tokens in a batch of size N . Similarly, denote

$$\mathbf{t}_k^{\text{mask}} = \mathbf{t}_k \setminus \mathbf{t}_k^{\text{unmask}} = [t_k^{(1)}, \dots, t_k^{(m')}] \in \mathbb{R}^{V \times m'}, k = 1, \dots, N \quad (4.12)$$

as the one-hot expression of the ground truth. Then the lyrics reconstruction loss of a mini-batch could be written as

$$\mathcal{L}_{r-t} = -\frac{1}{N} \sum_{k=1}^N \left[\frac{1}{m'} \sum_{i=1}^{m'} t_k^{(i)\top} \log \text{softmax}(\hat{t}_k^{(i)}) \right], \quad (4.13)$$

where the $\text{softmax}(\cdot)$ function was defined as

$$\text{softmax}([x_1, \dots, x_n]) = \left[\frac{\exp(x_1)}{\sum_{i=1}^n \exp(x_i)}, \dots, \frac{\exp(x_n)}{\sum_{i=1}^n \exp(x_i)} \right]. \quad (4.14)$$

The total reconstruction loss \mathcal{L}_r was a weighted sum of \mathcal{L}_{r-a} and \mathcal{L}_{r-t} :

$$\mathcal{L}_r = \mathcal{L}_{r-a} + \lambda_t \cdot \mathcal{L}_{r-t}. \quad (4.15)$$

4.2.2 Contrastive audio-lyrics learning

Our contrastive learning method took the other two passes to the joint encoder. The outputs of the modal-specific encoders, $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{t}}$ were passed into the joint encoder E_j , and mean-pooled along the sequence to become vectors.

$$\mathbf{c}_a = \text{MeanPool}(E_j(\tilde{\mathbf{a}}; \text{LN}_a)) \in \mathbb{R}^d, \quad (4.16)$$

$$\mathbf{c}_t = \text{MeanPool}(E_j(\tilde{\mathbf{t}}; \text{LN}_t)) \in \mathbb{R}^d \quad (4.17)$$

The contrastive loss was computed under the assumption that a pair of tokens $\{\mathbf{a}_i, \mathbf{t}_j\}$ were only considered positive if $i = j$, namely two modalities of the same sample, otherwise they were negative.

$$\mathcal{L}_c = -\frac{1}{N} \sum_{k=1}^N \log \left[\frac{\exp(\text{sim}(\mathbf{c}_{a_k}, \mathbf{c}_{t_k})/\tau)}{\sum_{j \neq k} \exp(\text{sim}(\mathbf{c}_{a_k}, \mathbf{c}_{t_j})/\tau) + \exp(\text{sim}(\mathbf{c}_{a_k}, \mathbf{c}_{t_k})/\tau)} \right]. \quad (4.18)$$

Here $\text{sim}(\cdot, \cdot)$ was the cosine similarity, and τ was the temperature.

The total loss of the self-supervised pretraining stage was a weighted sum of the contrastive and the reconstruction losses.

$$\mathcal{L} = \mathcal{L}_c + \lambda_r \cdot \mathcal{L}_r. \quad (4.19)$$

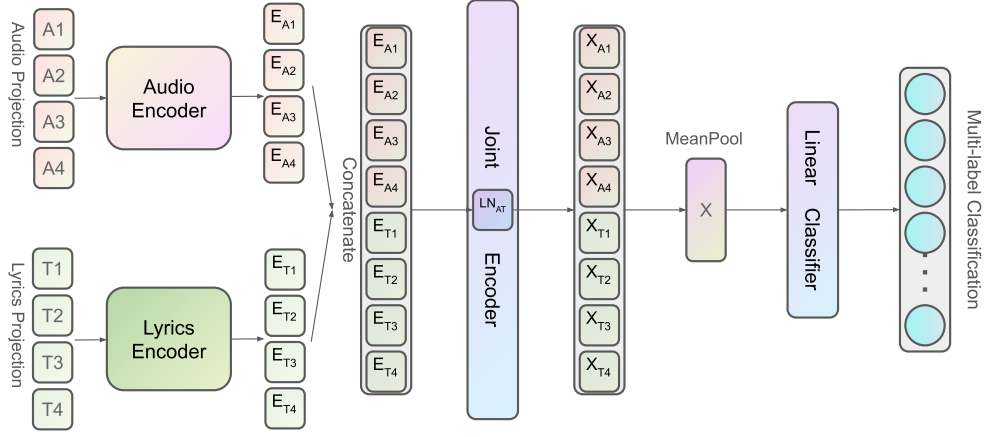


Figure 4.3: The model framework in the fine-tuning stage. No masking operation was performed, and the decoder was replaced by a linear classifier.

4.3 Fine-tuning and classification

In the fine-tuning stage, the joint decoder was discarded and only the encoders were kept (Figure 4.3). No masking operation was performed. The audio embeddings \mathbf{a} and the lyrics embeddings \mathbf{t} were added with a modality type embedding \mathbf{E}^M and a 2D sinusoidal position encoding \mathbf{E}^P , then fed into modal-specific encoders E_a and E_t , respectively.

$$\tilde{\mathbf{a}} = E_a \left(\mathbf{a} + \mathbf{E}_a^M + \mathbf{E}_a^P \right) \in \mathbb{R}^{d \times m}, \quad (4.20)$$

$$\tilde{\mathbf{t}} = E_t \left(\mathbf{t} + \mathbf{E}_t^M + \mathbf{E}_t^P \right) \in \mathbb{R}^{d \times n}. \quad (4.21)$$

A joint representation \mathbf{x} was acquired by passing the concatenation $[\tilde{\mathbf{a}}, \tilde{\mathbf{t}}]$ into the joint encoder and mean-pooling the output.

$$\mathbf{x} = \text{MeanPool} \left(E_j([\tilde{\mathbf{a}}, \tilde{\mathbf{t}}]; \text{LN}_{at}) \right) \in \mathbb{R}^d. \quad (4.22)$$

A multi-layer perceptron classifier $\text{MLP}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^C$ was attached to the joint encoder, where C was the number of classes in the music genre classification problem. In order to enable a multi-label classification, Sigmoid activation functions were utilized.

$$\hat{\mathbf{y}} = \text{MLP}(\mathbf{x}) \in \mathbb{R}^C. \quad (4.23)$$

Each digit of the output $\hat{\mathbf{y}}^{(c)} \in (0,1)$ represented the predicted probability of a class c ($c = 1, \dots, C$). The loss function we used was the binary cross-entropy (BCE) loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [\mathbf{y}_i^\top \quad \mathbf{1}^\top - \mathbf{y}_i^\top] \begin{bmatrix} \log \hat{\mathbf{y}}_i \\ \log(\mathbf{1} - \hat{\mathbf{y}}_i) \end{bmatrix}, \quad (4.24)$$

where $\mathbf{y} \in \mathbb{R}^C$ was a one-hot expression of the ground truth, and N referred to the batch size.

4.4 Motivation behind the methods

4.4.1 Why audio and lyrics

As we have introduced in Chapter 2, there are a variety of research works leveraging multimodal approaches to solve the music genre classification problem by combining audio and other modal-

ities, such as visual (album image and video), text (lyrics, user reviews and track metadata) and symbolic representations (MIDI, MusicXML and KERN). Among all these modalities, There were mainly two reasons of choosing lyrics in our multi-modal learning. Firstly, music audio and lyrics have significantly lower risk of mismatching compared to other modalities, since a part of lyrics are always audible in any segment of an audio. In contrast, taking an audio-visual dataset as an example, since not all the sound sources are visible and not all the visual objects make sound, it might contain a lot of mismatching or weakly matching modality pairs. This results in more false positive samples and lower quality of representations learnt in contrastive learning. Secondly, as argued in [Ala+20], audio modality is usually much more fine-grained than language (a few descriptive words or sentences), making them unsuitable to be projected into the same latent space. However, lyrics have a temporal correspondence with music audio thus they share a similar granularity.

4.4.2 Combining two self-supervised learning methods

[BAM18] proposed to divide multimodal learning into two categories: joint and coordinated representations. The former project unimodal signals into a common latent space, while the latter process them separately, but enforce certain similarity constraints on them. On one hand, many multimodal contrastive learning frameworks learn the coordinated representations by bringing paired modalities closer and separate mismatched ones further away in the latent space. On the other hand, we can extend the MAE, which leverages the masked data modeling, to multimodal scenarios, by mapping different modalities into the same latent space using a joint encoder, so as to learn the joint representations.

Although the two self-supervised learning methods are usually applied individually, they have pros and cons that may be complementary to each other. First, contrastive learning focuses on how to correctly pair the different modalities, thus reduces the misalignment when fusing them together, while losing some critical intra-modal information that is unique to each model. Masked data modeling forces the model to encode the unique information of each modality by recovering corrupt inputs, but inevitably lacks knowledge of intermodal correspondence and suffers from modality misalignment [Gon+23]. Second, contrastive learning is data hungry, requiring large dataset and large batch size to fully unlock its power, while masked data modeling works better under a low-data regime. Therefore, combining these two methods might be beneficial in our audio-text multi-modal approach to learn both coordinated and joint, intermodal and intramodal information.

4.4.3 Modality fusion

In this subsection, we are going to show that, not only did the joint encoder E_j serve as a self-attention module to learn the intra-modal information, but it could also be seen as a cross-modal attention which encoded the inter-modal relationship and dynamically assigned stronger weights to modalities with greater importance.

For the sake of simplicity, we only need to figure out what happened in the basic component – the scaled dot-product attention. Denote the concatenation of the outputs of two modal-specific encoders as

$$X = \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{t}} \end{bmatrix} \in \mathbb{R}^{(n+m) \times d}. \quad (4.25)$$

The query, key and value matrices were formed as

$$Q_X = XW_Q = \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{t}} \end{bmatrix} W_Q = \begin{bmatrix} \tilde{\mathbf{a}}W_Q \\ \tilde{\mathbf{t}}W_Q \end{bmatrix} = \begin{bmatrix} Q_a \\ Q_t \end{bmatrix} \in \mathbb{R}^{(n+m) \times d_k}, \quad (4.26)$$

$$K_X = XW_K = \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{t}} \end{bmatrix} W_K = \begin{bmatrix} \tilde{\mathbf{a}}W_K \\ \tilde{\mathbf{t}}W_K \end{bmatrix} = \begin{bmatrix} K_a \\ K_t \end{bmatrix} \in \mathbb{R}^{(n+m) \times d_k}, \quad (4.27)$$

$$V_X = XW_V = \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{t}} \end{bmatrix} W_V = \begin{bmatrix} \tilde{\mathbf{a}}W_V \\ \tilde{\mathbf{t}}W_V \end{bmatrix} = \begin{bmatrix} V_a \\ V_t \end{bmatrix} \in \mathbb{R}^{(n+m) \times d_v}. \quad (4.28)$$

When E_j took X as its input, the scaled dot-product attention was computed as

$$\text{Attention}(Q_X, K_X, V_X) = \text{softmax} \left(\frac{Q_X K_X^T}{\sqrt{d_k}} \right) V_X. \quad (4.29)$$

With a little bit of approximation, we would be able to show that this formula contained both self-attention and cross-modal attention.

$$\text{Attention}(Q_X, K_X, V_X) = \text{softmax} \left(\frac{Q_X K_X^T}{\sqrt{d_k}} \right) V_X \quad (4.30)$$

$$= \text{softmax} \left(\frac{1}{\sqrt{d_k}} \begin{bmatrix} Q_a \\ Q_t \end{bmatrix} \begin{bmatrix} K_a^T & K_t^T \end{bmatrix} \right) \begin{bmatrix} V_a \\ V_t \end{bmatrix} \quad (4.31)$$

$$= \text{softmax} \left(\begin{bmatrix} \frac{1}{\sqrt{d_k}} Q_a K_a^T & \frac{1}{\sqrt{d_k}} Q_a K_t^T \\ \frac{1}{\sqrt{d_k}} Q_t K_a^T & \frac{1}{\sqrt{d_k}} Q_t K_t^T \end{bmatrix} \right) \begin{bmatrix} V_a \\ V_t \end{bmatrix} \quad (4.32)$$

$$\simeq \begin{bmatrix} \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q_a K_a^T \right) & \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q_a K_t^T \right) \\ \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q_t K_a^T \right) & \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q_t K_t^T \right) \end{bmatrix} \begin{bmatrix} V_a \\ V_t \end{bmatrix} \quad (4.33)$$

$$= \begin{bmatrix} \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q_a K_a^T \right) V_a + \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q_a K_t^T \right) V_t \\ \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q_t K_a^T \right) V_a + \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q_t K_t^T \right) V_t \end{bmatrix} \quad (4.34)$$

$$= \begin{bmatrix} \text{Attention}(Q_a, K_a, V_a) + \text{Attention}(Q_a, K_t, V_t) \\ \text{Attention}(Q_t, K_a, V_a) + \text{Attention}(Q_t, K_t, V_t) \end{bmatrix}. \quad (4.35)$$

It was apparent that the scaled dot-product attention $\text{Attention}(Q_X, K_X, V_X)$ contained self-attention of both modalities: $\text{Attention}(Q_a, K_a, V_a)$ and $\text{Attention}(Q_t, K_t, V_t)$, as well as the cross-modal attention from audio to lyrics: $\text{Attention}(Q_a, K_t, V_t)$ and that from lyrics to audio: $\text{Attention}(Q_t, K_a, V_a)$.

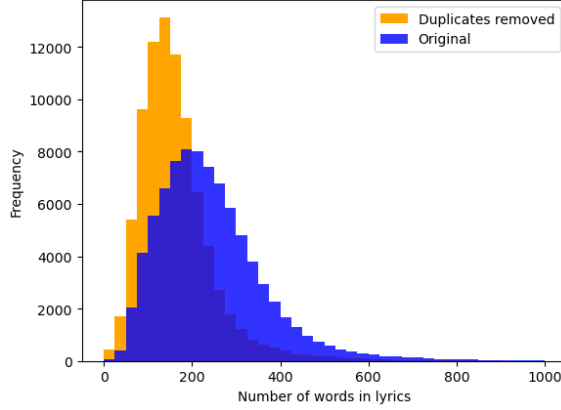


Figure 5.2: The distribution of the lyrics length before and after preprocessing. Both mean and variance decreased.

To explore the scalability of self-supervised pretraining, we created another dataset of music audio and lyrics, by simply computing an intersection of two existing datasets containing audio and lyrics, respectively. The music audio dataset was a proprietary dataset (provided by Dr. Igor Vatulkin) of 784,203 songs from the Million Song dataset [Ber+11], and the lyrics dataset was Genius Song Lyrics [Nay22], comprising 5,283,576 songs with lyrics. We computed their intersection based on an exact match of song title and artist name(s), restricted the language of lyrics to English only and preprocessed the lyrics, resulting in a dataset of size 113,589 (denoted as msd).

5.2 Experimental settings

5.2.1 Hyperparameters

In the pretraining stage, we used the whole m4a dataset. We trained the model for 50 epochs. The optimizer was Adam (weight decay $5e-7$, $(\beta_1, \beta_2) = (0.95, 0.999)$). The backbone learning rate was $5e-5$, which started to decay from the 10th epoch, with a rate of 0.5 every 10 epochs. The default batch size was 48. Generally speaking, 30 seconds of audio is redundant for most MIR tasks, including music genre classification. Thus, we started our experiments with short segments of 5 seconds' length, where we randomly cropped the audio to the given length and paired it with complete lyrics during training. In this way, we also augmented the data without much sacrifice of diversity. The audio was then converted to 128-dimensional log Mel filterbank (fbank) features, computed with a 25ms Hanning window and a 10ms hop size (we adopted this setting from CAV-MAE). Every segment of 5 seconds was sampled and padded to a length of 512 samples along the temporal dimension. For lyrics, we truncated or padded them to 256 tokens. The lyrics reconstruction loss had a weight of $\lambda_t = 1000$ because we found that the cross-entropy loss was much smaller than the MSE loss utilized by the audio reconstruction. The weights of the whole MDM loss is $\lambda_r = 1$. The temperature τ in the contrastive loss (Equation 4.18) was as small as 0.05, in order to ensure a sharp discrimination between the positives and the negatives. In fine-tuning, we performed a 5-fold cross-validation on the m4a dataset and trained the model for 10 epochs. The optimizer was the same Adam as in the pretraining, as was the batch size. We adopted a smaller learning rate of $1e-5$, which started to decay from the second epoch, with a rate of 0.5 every 4 epochs.

Regarding the details of our model, all encoders were 768-dimensional and had 12 attention heads, and the joint decoder was 512-dimensional and had 16 attention heads. Two modal-specific encoders, E_a and E_t , had 5 layers, and the joint encoder E_j was one-layer. The joint decoder D_j had 4 layers. Such a model had 130 million trainable parameters in total. The tokenizer we used was that of the BERT-base-uncased [Dev+19], with a vocabulary size of 30522. To compensate for the lack of large scale pretraining datasets, we used the pretrained weights of CAV-MAE [Gon+23]

for most parts of the model, except the lyrics encoder E_t , where we used the weights of the first 5 layers of BERT-base-uncased.

It took about 15 hours on average to run the pretraining, and another 15 hours to fine-tune (5 folds) on a single NVIDIA H100 Tensor Core GPU 80GB.

5.2.2 Metrics

The genres in our dataset m4a had a multi-label nature, thus we had to employ metrics designed for multi-label classification. Given a data point x_i , the set of its labels Y_i and the corresponding prediction of the model \hat{Y}_i , metrics for single-label multi-class classification only measures the exact match between the prediction and the ground truth, i.e., $Y_i = \hat{Y}_i$. However, an exact match of two label sets is too rigorous for multi-label classification, where partial correctness, namely the intersection of the ground truth and the prediction $Y_i \cap \hat{Y}_i$, also matters. Therefore, we extended some common metrics (accuracy, precision, recall and F-measure) in single-label classification to the multi-label classification scenario.

Formally, we denoted $D = \{(X_i, Y_i) | i = 1, 2, \dots, N, Y_i = \{0, 1\}^L\}$ as the dataset, where Y_i was the ground-truth label, \hat{Y}_i was the model prediction, and $L = \{l_1, \dots, l_C\}$ was the set of all labels, then

$$\text{Accuracy } Acc = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|} \quad (5.1)$$

$$\text{Precision } P = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap \hat{Y}_i|}{|\hat{Y}_i|} \quad (5.2)$$

$$\text{Recall } R = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap \hat{Y}_i|}{|Y_i|} \quad (5.3)$$

$$F_1 = \frac{1}{N} \sum_{i=1}^N \frac{2|Y_i \cap \hat{Y}_i|}{|Y_i| + |\hat{Y}_i|}. \quad (5.4)$$

To generate the predicted label $\hat{Y}_i = \{0, 1\}^L$ from the model output $Z_i \in \mathbb{R}^L$, we set the threshold to $\theta = \text{Sigmoid}(-1.5)$, i.e., $\hat{Y}_i = \mathbb{I}(Z_i > \text{Sigmoid}(-1.5))$.

Another metric we utilized was the mean average precision (mAP). Note that the definition of mAP differs slightly in different research and implementations. Here, we defined mAP as the mean value of the average precisions (APs) over all label classes.

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^C \text{AP}_c. \quad (5.5)$$

Denote the model output of the dataset D as $Z = [z^1, \dots, z^C] \in \mathbb{R}^{N \times C}$, then the predicted values of label l_c was $z^c = [z_1^c, \dots, z_N^c]^T$, which could be sorted as $z_{(1)}^c \geq \dots \geq z_{(N)}^c$. The n -th threshold was set to $\theta_n = z_{(n)}^c$, and the n -th precision P_n^c and recall R_n^c were computed on the n -th prediction of label l_c :

$$\hat{y}^c = \mathbb{I}(z^c > \theta_n). \quad (5.6)$$

The average precision (AP) was a weighted average of precisions along the precision-recall curve:

$$\text{AP}_c = \sum_{n=1}^N (R_n^c - R_{n-1}^c) P_n^c, \quad R_0^c = 0. \quad (5.7)$$

In addition, to measure the importance of modalities (audio and lyrics) on each genre class, we also defined a new metric which was similar to the permutation feature importance [FRD19] in mechanism. Specifically, the importance of modality was the difference between the multimodal AP and the unimodal AP where that modality was missing, which could also be seen as the loss in AP. The larger the difference (loss), the more important the modality. Because the average precision

sometimes varied greatly between different genres, we also divided it by the original multimodal AP to calculate a percentage.

$$I_c^A = \frac{AP_c^{MM} - AP_c^L}{AP_c^{MM}}, \quad (5.8)$$

$$I_c^L = \frac{AP_c^{MM} - AP_c^A}{AP_c^{MM}}, \quad (5.9)$$

where I_c^A , I_c^L were the importance metrics of audio and lyrics, and AP_c^{MM} , AP_c^A , AP_c^L referred to the average precision of the multimodal, audio-only and lyrics-only approaches, respectively. Moreover, for the sake of readability, we aggregated the 50 genre classes into 13 coarse genres (hip hop, experimental, blues, soundtrack, folk, metal, rock, punk, pop, r&b & soul, country, electronic and jazz) based on the categorization of music genres and subgenres in Wikipedia [con24], and averaged their importance metrics inside each coarse genre.

5.3 Experimental design

- **The ablation study of modalities.** To find out whether combining audio and lyrics was complementary in music genre classification, we pretrained and fine-tuned the model with a single modality, and saw how it impacted the model performance. Additionally, we conducted one-sided, pairwise Wilcoxon signed-rank tests to test whether the improvement of multimodal approach on each of the three metrics, accuracy, F_1 and mAP, is statistically significant. The reason we used a non-parametric Wilcoxon test instead of a more efficient but parametric T test was the extremely small number of samples (5-fold), which made it hard to validate the normality of data. The null and alternative hypotheses were as follows:
 - H_0 : Combining audio and lyrics in music genre classification did not improve the model performance compared to the unimodal approach using audio/lyrics.
 - H_1 : Combining audio and lyrics in music genre classification did improve the model performance compared to the unimodal approach using audio/lyrics.
- **The ablation study of self-supervised learning methods.** Similar to the previous study, we ran the experiment with a partial loss function: contrastive only and MDM only, as well as a naive approach of no self-supervised pretraining at all. We also conducted Wilcoxon signed-rank tests with the following hypotheses:
 - H_0 : Using contrastive learning and MDM altogether in the multimodal audio-lyrics music genre classification did not increase the model performance compared to using either/neither of them.
 - H_1 : Using contrastive learning and MDM altogether in the multimodal audio-lyrics music genre classification did increase the model performance compared to using either/neither of them.
- **Comparing different lengths of audio and lyrics.** As we have discussed, 30 seconds is superfluous for music genre classification, but we would also like to know whether we could use a even shorter length (5 or 2.5 seconds) to make the model focus on local patterns with more details and improve the computational efficiency. However, audio segments that were too short might lead to weak matching between audio and lyrics (since the audio contain too few words) and undermine the quality of contrastive learning. For lyrics, 256 tokens was still a large number for the model input. Due to the limited computational resources, a shorter input sequence (e.g., 64 or 128 tokens) without losing critical information was also desired.
- **More pretraining data.** To explore the scalability of self-supervised pretraining, we compared the model pretrained on only m4a and on both m4a and msd.

- **Comparing different model sizes** Our proposed model was a large one. Therefore, it was essential to explore whether down-scaling the model parameter size could be an acceptable trade-off between model performance and training costs. Moreover, the m4a dataset we used for pretraining was not a large one (around 80K), compared to other audio pretraining datasets, such as Audioset-2M. To prevent overfitting, it might be beneficial to choose a model with less parameters.
- **Comparing different batch sizes in contrastive learning.** As discussed in Section 2, negative-sampling-based contrastive learning depends on large batch size to ensure sufficient negative samples, in order to create a more challenging learning environment. Therefore, we experimented with both smaller (16) and larger (144) batch sizes to uncover the impact of contrastive batch size.
- **Comparing different masking ratios.** MDM first achieved success in natural language processing with a rather low masking ratio, e.g., 15% for BERT. However, due to the signal and informational sparsity, computer vision models have to utilize a much higher masking ratio for images, e.g., 75% for MAE. Audio spectrogram is similar to images, thus a high masking ratio is also applied, such as Audio-MAE (75%), CAV-MAE (65% to 75%). Furthermore, under a multimodal setting which combines textual data with other modalities, researchers found that text must also be masked with a high masking ratio. For example, M3AE found that a 75% masking ratio was optimal for multimodal visual-language learning. Since there are not many works on audio-language learning, it is of great interest to find out whether we should also adopt a high masking ratio for both of the two modalities, or more specifically, for music audio and lyrics. Furthermore, we gave some examples of audio spectrogram inpainting under different masking ratios to see their impacts on the inpainting ability.
- **Comparing our model with other baselines.** After considering the existing research on (multimodal) music genre classification, we implemented two unimodal and one multimodal baselines. For audio spectrograms, we used Inception-v4 [Sze+17] with ImageNet [Den+09] pretrained weights, a CNN with ResNet-like residual connections. For lyrics, we simply chose BERT-base-uncased. Our multimodal baseline was a combination of the two aforementioned models, using them as modal-specific encoders and a single-layer, 768-dimensional transformer encoder with 12 attention heads to fuse the outputs of the modal-specific encoders. We called this model ALNet. It was trained with a loss function which was a weighted sum of contrastive and classification loss.

$$\mathcal{L} = \mathcal{L}_{\text{classification}} + 0.05 \cdot \mathcal{L}_{\text{contrastive}}. \quad (5.10)$$

Chapter 6

Results

In this chapter, we are going to display the results of experiments proposed in the last chapter, including the ablation study of modalities and self-supervised learning methods, as well as tuning various hyperparameters. We also provide detailed interpretation and analysis of these results, which leads to the answer of the research questions.

6.1 Ablation of modalities

Metrics	Ours A-L	Modality Ablation A	L
Acc	33.48 ± 0.19	30.36 ^(*) ± 0.23	23.11 ^(*) ± 0.21
F_1	43.21 ± 0.23	39.33 ^(*) ± 0.26	30.48 ^(*) ± 0.34
mAP	30.94 ± 0.27	24.55 ^(*) ± 0.21	15.90 ^(*) ± 0.17

Table 6.1: Results of the modality ablation study. "A" referred to audio and "L" referred to lyrics. All the results were averaged on 5-fold cross validation, presented with mean and standard deviation values. The p-values of the statistical tests were marked by the number of asterisks: (*)- $p < 0.05$, (**) - $p < 0.01$, (***) - $p < 0.001$.

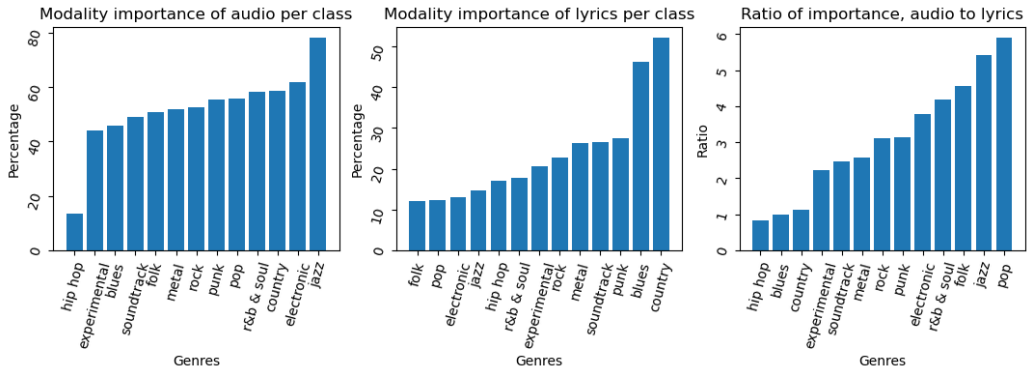


Figure 6.1: The modality importances of audio and lyrics each (coarse) genre. We also calculated the ratio of audio importance to lyrics importance.

From Table 6.1, we could see that our multimodal approach largely outperformed the unimodal ones with audio or lyrics alone in all three metrics with statistical significance ($0.01 < p < 0.05$). Two modalities compensated for the missing information of each other, and together enhanced the representation with the distinct features they captured respectively, thus resulted in the complementarity of the multimodal approach. More specifically, the audio-only model was much better

than the lyrics-only model. A possible reason was the complexity of lyrics. In many masked language modeling pretraining scenarios, the texts are usually short and descriptive, in a simple form and in only a few sentences (e.g., captions of an image or a segment of audio). However, lyrics are longer, freer in its form and more unpredictable, containing incomplete sentences, colloquial expressions such as abbreviations and slangs. In this sense, learning the noisy lyrics effectively remains a challenge, while learning audio for music classification is much more simple and robust, where we can acquire satisfying results with a CNN on spectrograms.

Figure 6.1 told us that audio and lyrics were of different importances to different genres. Most genres (11 out of 13) relied on audio to a very similar extent, while hip hop and jazz were the outliers on two extremes (low and high respectively). When it came to lyrics, they were far more important to blues and country music than other genres. Moreover, it was noteworthy that audio was in general more important than lyrics for almost all the genres (ratio greater than 1). For some genres, such as jazz, r&b & soul, electronic, rock and punk, audio was much more helpful than lyrics in classifying the genre, probably because of their distinctive sound effects (rhythm, timbre, instruments, etc.). On the other hand, for those genres that were very characteristic in both sound effect and the writing style of lyrics – hip hop, blues and country, audio and lyrics were of almost equal importance. It was also noteworthy that higher importance of one modality did not necessarily imply a lower importance of the other, since we measured the importance through the loss of AP. In the experiment, the multimodal prediction of each sample were hardly a union of the two unimodal predictions, thus it was difficult to find any valid relationships between their APs. Sometimes, we observed that the audio-only and lyrics-only models produced very similar predictions, thus they were almost equally good; in some other cases, the model benefited from the multimodal setting and generated correct answers unseen in any of the two unimodal predictions. The mechanism behind these results remained rather unexplainable.

6.2 Ablation of self-supervised learning methods

Metrics	Ours	Contrastive	MDM	No SSL
Acc	33.48 ± 0.19	31.81 ^(*) ± 0.21	32.86 ^(*) ± 0.32	29.76 ^(*) ± 0.21
F_1	43.21 ± 0.23	41.81 ^(*) ± 0.22	43.08 ⁽⁻⁾ ± 0.12	39.27 ^(*) ± 0.12
mAP	30.94 ± 0.27	28.91 ^(*) ± 0.20	30.88 ⁽⁻⁾ ± 0.42	25.41 ^(*) ± 0.34

Table 6.2: Results of the self-supervised learning ablation study. The p-values of the statistical tests were marked by the number of asterisks: (*)- $p < 0.05$, (**) - $p < 0.01$, (***) - $p < 0.001$. (–) meant that there was no statistical significance.

See Table 6.2. It turned out that combining two self-supervised learning methods resulted in a significant ($0.01 < p < 0.05$) increase of performance, compared to applying only one of them, though the improvement was minor. However, the increase of performance resulting from applying two methods respectively was still remarkable compared to the naive method with no self-supervised pretraining. It was also noteworthy that MDM-only method was 1% to 2% higher than the contrastive-only one in all three metrics. The differences in F_1 and mAP between the MDM-only and multimodal approach had no statistical significance ($p > 0.05$). Moreover, considering the low contrastive accuracies in Table 6.3 and 6.4, it should not be overlooked that the effect of contrastive learning is below our expectation. We believed that there were mainly two reasons contrastive learning is inferior to MDM in our experiments. Firstly, as we just discussed in the last question, lyrics are noisier, more informative and more complicated than many other texts, and the corresponding relationship of music audio and lyrics is on a fine-grained level of words and syllables along the temporal dimension. Although audio and lyrics do have lower risk of mismatching in contrastive learning, as discussed in Section 4.4, such an intricate relationship requires much extra effort to learn, compared to descriptive texts which contains mostly high-level semantics like an object or an event. Moreover, as we concluded in Chapter 2.2, contrastive learning mostly fo-

cuses on global information, while the point-wise nature determines that MDM models mostly low-level, word-to-word semantics. Thus, no wonder the latter performed better in our experiments. Secondly, contrastive learning is more data-hungry than MDM, while MDM works better under a low-data regime. Classic contrastive learning frameworks based on negative-sampling are trained on very large datasets (SimCLR: 1M, MoCo: 1M and 1B). In audio representation learning, one of the most common pretraining datasets is the AudioSet-2M [Gem+17]. However, our datasets for pretraining were only of size 80K and 190K, around 1/10 of the aforementioned ones. Therefore, a lack of large-scale pretraining dataset might also be a reason that contrastive learning behaves not as well as MDM.

6.3 Hyperparameter tuning

Metrics	10 seconds	5 seconds	2.5 seconds
Acc	32.05 \pm 0.31	33.48 \pm 0.19	31.43 \pm 0.30
F_1	42.16 \pm 0.23	43.21 \pm 0.23	40.86 \pm 0.35
mAP	29.13 \pm 0.38	30.94 \pm 0.27	27.35 \pm 0.43
CL Acc	13.40	16.67	14.52

Table 6.3: Results of comparing different audio lengths. We also recorded the highest contrastive accuracy in the pretraining stage, which was the ratio of cases in which a sample was more similar to its positive counterpart than any other negative ones.

- **Comparing different length of audio.** See Table 6.3. 5 seconds of audio was better for the model performance compared to 10 and 2.5 seconds. Note that 5-second audio also has the highest contrastive accuracy. Nevertheless, all the contrastive accuracies were rather low ($< 20\%$). If the input length of audio was too long, it would be very resource-intensive and too noisy for the model to learn properly. Conversely, if the input was too short, it might lead to weak matching between audio and lyrics, since the audio contained too few words. Therefore, we could see that the intermediate 5-second was the best choice for the sake of contrastive learning and overall model performance.

Metrics	256	128	64
Acc	33.48 \pm 0.19	30.10 \pm 0.34	29.67 \pm 0.20
F_1	43.21 \pm 0.23	39.23 \pm 0.66	38.71 \pm 0.19
mAP	30.94 \pm 0.27	25.58 \pm 0.20	24.97 \pm 0.12
CL Acc	16.67	13.23	10.37

Table 6.4: Result of comparing different lyrics length in number of tokens. CL Acc was the highest contrastive accuracy recorded in the pretraining stage.

- **Comparing different length of lyrics.** See Table 6.4. As of the input length of lyrics, we also wanted it to be as short as possible to increase the efficiency, while the results showed that a longer input sequence of lyrics with 256 word tokens was better than 128 and 64. Note that the 256-token model also had the highest contrastive accuracy. It was probably because truncating lyrics increased the risk of mismatching – the already very short audio segments completely missed that part of lyrics, which we could also tell from the contrastive accuracies.

Metrics	m4a	m4a + msd
Acc	33.48 ± 0.19	33.38 ⁽⁻⁾ ± 0.17
F_1	43.21 ± 0.23	43.61 ^(*) ± 0.17
mAP	30.94 ± 0.27	31.60 ^(*) ± 0.31

Table 6.5: Results of using different pretraining datasets: pretraining on our original dataset m4a, and on m4a + msd together. The p-values of the statistical tests were marked by the number of asterisks: (*)- $p < 0.05$, (**) - $p < 0.01$, (***) - $p < 0.001$. (-) meant that there was no statistical significance ($p > 0.05$).

- **More pretraining data.** See Table 6.5. Using a larger pretraining dataset was a way to reduce overfitting. Thus, combining two pretraining datasets, m4a and msd, indeed resulted in an increase in two metrics: F_1 and mAP, which was marginal but with statistical significance. A possible reason could be the relatively limited size of our enlarged dataset, which was of size 190K compared to 80K of the original m4a dataset. It would be of great interest to explore the scalability of our model, when real large-scale data of over 1M is available.

Metrics	(11,8)	(5,4)	(2,2)
Acc	32.89 ± 0.30	33.48 ± 0.19	32.90 ± 0.26
F_1	42.30 ± 0.33	43.21 ± 0.23	42.48 ± 0.26
mAP	29.85 ± 0.24	30.94 ± 0.27	30.05 ± 0.14

Table 6.6: Results of comparing different model sizes (numbers of hidden layers in each modality-specific encoder and the joint decoder).

- **Comparing different model sizes.** See Table 6.6. Since our dataset was rather small, it was sensible to also down-scale the size of our model, in order to prevent overfitting. The results demonstrated that a medium size of model with modality-specific encoders with 5 layers and a decoder with 4 layers was the best choice. However, a light-weight model with only 2 layers of modality-specific encoders and 2 layers of the decoder could yield comparable results to our optimal choice, and even better than the large-size model, which made a good trade-off between model performance and computational costs.

Metrics	144	48	16
Acc	33.54 ± 0.24	33.48 ± 0.19	32.32 ± 0.16
F_1	43.18 ± 0.17	43.21 ± 0.23	41.92 ± 0.16
mAP	30.85 ± 0.26	30.94 ± 0.27	29.19 ± 0.34

Table 6.7: Results of comparing different batch sizes in contrastive learning.

- **Comparing different batch sizes in contrastive learning.** See Table 6.7. We could see that increasing the batch size from 16 to 48 resulted in an increase in all three metrics, while a further increase from 48 to 144 did not make much difference. As we previously discussed in Section 2.2, negative-sampling based contrastive learning requires large batch size to create a challenging and diverse learning environment, while there is also an important trade-off between the effect of contrastive learning, and the computational costs. Indeed, small batch size like 16 was significantly inferior to larger ones, but the gain of performance by further tripling the batch size from 48 to 144 was merely marginal. However, it was noteworthy that 144 was still far from the settings adopted by some successful CL frameworks, e.g., 4096 for SimCLR, thus there could be a remarkable improvement if we manage to increase the batch size drastically, along with a much larger dataset. Meanwhile, that has to be run on a number of state-of-the-art GPUs, which is hard to realize in a university laboratory.

Metrics	(0.3, 0.3)	(0.5, 0.5)	(0.65, 0.65)	(0.75, 0.75)	(0.85, 0.85)
Acc	32.92 \pm 0.20	33.57 \pm 0.13	33.33 \pm 0.09	33.48 \pm 0.19	32.31 \pm 0.11
F_1	43.09 \pm 0.16	43.86 \pm 0.09	43.61 \pm 0.11	43.21 \pm 0.23	42.43 \pm 0.10
mAP	30.78 \pm 0.28	32.07 \pm 0.24	31.37 \pm 0.42	30.94 \pm 0.27	29.78 \pm 0.34
CL Acc	26.74	33.22	23.08	16.67	11.63

Table 6.8: Results of comparing different masking ratios of (audio, lyrics). CL Acc was the highest contrastive accuracy recorded in the pretraining stage.

- **Comparing different masking ratios.** See Table 6.8. We tested a series of choices from 0.3 to 0.85, and found the model performance first increased, then decreased as the masking ratio went up. The optimal ratio was 50% for both audio and lyrics. The differences caused by varying masking ratios were not significantly, especially for ratios between 50% and 75%. Usually, a higher masking ratio leads to a challenging learning task, forcing the model to learn more about the inherent structure in data. What was interesting in the results was that the optimal masking ratio of audio and lyrics was a little lower than the findings of some similar research, such as the audio-visual CAV-MAE (65% to 75%) and the visual-language M3AE (75%). We believed that it was due to the difficulty of learning lyrics. Moreover, the masking ratio also had an impact on contrastive learning. Lower masking ratios mitigated the difficulty of matching the partially masked audio and lyrics, thus resulted in higher contrastive accuracies.

Metrics	Ours A-L	Inception v4 A	BERT L	ALNet A-L
Acc	33.48 \pm 0.19	34.33 \pm 0.48	27.94 \pm 0.14	35.58 \pm 0.32
F_1	43.21 \pm 0.23	44.08 \pm 0.52	35.67 \pm 0.16	45.27 \pm 0.42
mAP	30.94 \pm 0.27	31.03 \pm 0.37	20.34 \pm 0.08	32.82 \pm 0.71

Table 6.9: Results of comparing our model to three other baselines: the unimodal Inception v4 (audio) and BERT (lyrics), as well as the multimodal ALNet.

- **Comparing our model with other baselines.** See Table 6.9. Unfortunately, our model was outperformed by a better unimodal CNN working on audio spectrograms, though the results were close. On the other hand, CNN had higher metrics compared to the other unimodal baseline BERT which operated on lyrics. Moreover, the ALNet also benefited from a multimodal approach combining Inception v4 and BERT. The reason why a CNN could outperform our model was that the unimodal backbones of our model were inferior to the baselines in Table 6.9. Our lyrics encoder was of similar architecture to BERT-base-uncased but had only 6 layers, while the latter had 12. In fact, we had also experimented with a 12-layer setting, and the results were very close to those of BERT, implying that it was probably just underfitting in the unimodal scenario. Therefore, the key problem lied in the difference between the transformer-based MAE architecture of the audio encoders and the traditional convolutional neural network in the baselines. Many research papers have already confirmed that ViTs perform worse than CNNs when trained on small datasets [Dos20; Rag+21; Tou+21]. CNNs are based on strong assumptions called inductive biases, like spatial locality and translation invariance. These assumptions help CNNs excel at many tasks, especially on smaller datasets where there is less data to learn from. In contrast, the transformer model does have such biases. On one hand, transformers need a lot of data to learn meaningful patterns from scratch, without the help of biases. On the other hand, transformers are more flexible because they rely on self-attention mechanisms that can model both local and global dependencies in the data, without being constrained by the spatial biases inherent in CNNs. This flexibility allows them to capture richer, more complex relationships in the data. This is why transformers tend to outperform CNNs in large-scale tasks when enough data are available. Moreover, there is no better choice than applying MDM on audio spectrogram data with a

transformer-based model such as MAE, since we need to learn the contextual relationship between spectrogram tokens. Therefore, despite falling short of some metrics, we stuck to the more complicated transformers as our model backbones, since they implied far more possibilities and higher potential in the future.



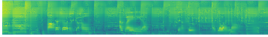

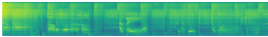

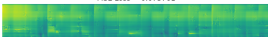
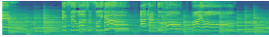

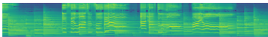

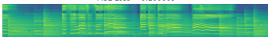

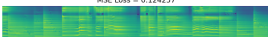


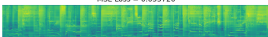
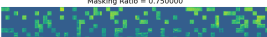
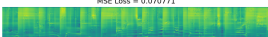
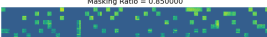
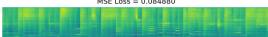
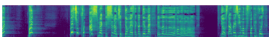

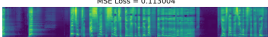
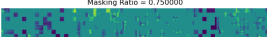
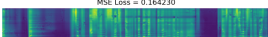
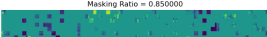
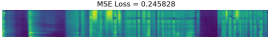
Original spectrograms	Masked spectrograms	Reconstructed spectrograms
	Masking Ratio = 0.500000 	MSE Loss = 0.050291 
	Masking Ratio = 0.750000 	MSE Loss = 0.062612 
	Masking Ratio = 0.850000 	MSE Loss = 0.073761 
	Masking Ratio = 0.500000 	MSE Loss = 0.072728 
	Masking Ratio = 0.750000 	MSE Loss = 0.100609 
	Masking Ratio = 0.850000 	MSE Loss = 0.124257 
	Masking Ratio = 0.500000 	MSE Loss = 0.055720 
	Masking Ratio = 0.750000 	MSE Loss = 0.070771 
	Masking Ratio = 0.850000 	MSE Loss = 0.084880 
	Masking Ratio = 0.500000 	MSE Loss = 0.113004 
	Masking Ratio = 0.750000 	MSE Loss = 0.164230 
	Masking Ratio = 0.850000 	MSE Loss = 0.245828 

Table 6.10: Results of audio spectrogram inpainting under different masking ratios: 0.5, 0.75 and 0.85.

- **Audio spectrogram inpainting.** See Tabel 6.10. The lower the masking ratio, the better the effect of inpainting, with lower MSE loss values. Even when 85% of patches were masked, the model was still able to reconstruct the data to a great extent and learn the overall structure correctly. For lower masking ratio like 0.5, many fine-grained details were successfully recovered by the model. Additionally, we do not include the results of lyrics inpainting (masked token prediction) because the model could only fill in some common stop words like “I”, “you”, “the” and punctuations in most cases, even if the masking ratio was as low as 15%.

Chapter 7

Discussion

We conclude our findings by answering the research questions raised in Chapter 1. Moreover, we also discuss the limitations of our work and possible directions of future research.

7.1 Conclusion

RQ 1 Is it complementary to utilize a multimodal learning method with audio and lyrics in music genre classification, compared to unimodal method based on audio or lyrics alone? Moreover, is audio more important than lyrics in music genre classification, or vice versa?

Our experiments showed that a multimodal audio-lyrics model was significantly better than its unimodal counterparts. The comparison with baselines also verified this conclusion: the ALNet model combining Inception v4 and BERT outperformed either of them. This was mainly because of two reasons. Firstly, different modalities capture different features of the same object. For example, in music, audio captures melody, rhythm, and timbre, while lyrics provide semantic and emotional context. Combining both allows the model to form a more complete and nuanced representation of music. Secondly, certain information may be missing or less prominent in one modality but available in another. For instance, in noisy or low-quality audio data, the lyrics might still convey useful information for genre classification. We also observed in the ablation study that an audio-only method performed better than its lyrics-only counterpart. We believed that it was because the lyrics were noisier, more complex and more unpredictable to learn than audio. More specifically, audio and lyrics showed different importances to different genres, while we found that in almost all the genres, audio was more important than lyrics.

RQ 2 Does self-supervised learning enhance the performance of our model? Does contrastive learning and masked data modeling perform well when applied respectively, and is it more helpful to combine them together?

Contrastive learning and MDM each contributed to the performance boost, with MDM showing a slight edge over contrastive learning. When combined, they offered even more complementary advantages. However, contrastive learning did not perform as expected, with accuracy below 20% in audio-lyrics matching. This underperformance likely stemmed from the fact that the lyrics are noisier and more complex than typical text, and audio-lyrics relationships were very intricate and fine-grained, requiring word-level precision that MDM could better capture. Additionally, contrastive learning generally required large datasets for effective training, unlike MDM, which performed better with limited data. Our smaller pretraining dataset likely further limited contrastive learning’s impact, highlighting MDM’s advantages under these conditions.

RQ 3 There are various hyperparameters in the proposed method, such as the input length of audio and lyrics, size of model and datasets, batch size in contrastive learning as well as masking ratio in masked data modeling. How do they affect the model performance?

As for the input length of audio and lyrics, we believed that it was important to make a good trade-off between the computational efficiency and the quality of contrastive learning. Thus, we found that an intermediate choice of 5 seconds was optimal for audio, while the longest 256 tokens were the best for lyrics. To reduce overfitting, we explored a downscaling of the model and an upscaling of the dataset. Even when there were only 2 layers in each modality-specific encoder and

2 layers in the decoder, the results were still comparable to the best one but with less computational costs. Meanwhile, increasing the pretraining data from 80K to 190K resulted in an (incremental) improvement of model performance. When it came to contrastive learning, a large batch size was vital to ensure enough negative samples and thus a good representation for downstream tasks. Nonetheless, a batch size of 48 was good enough compared to 16, while increasing it to 144 did not bring further benefits to the model. Lastly, we found that masking 50% of tokens for both audio and lyrics was optimal, which was lower than the settings in similar research (around 65% to 75%).

7.2 Limitations

Although this thesis accomplished success in finding the complementarity of a multimodal approach in music genre classification and combining different self-supervised learning methods, we admit there are some limitations. Firstly, our model was outperformed by a more simple, unimodal CNN, which implied that our proposed method might not be the most efficient under limited data. Furthermore, there might be other underlying reasons, and we did not manage to exhaust all possible solutions, e.g., trying other model architectures other than MAE. Secondly, in Section 4.4, we reckoned that audio and lyrics were a suitable combination for multimodal music genre classification, but the theory failed to explain our experimental results. It turned out that lyrics were more complex to learn than we thought, and sometimes the model might not be able to extract high-level semantics from the noisy lyrics.

7.3 Future research

A regret of our research was that our model did not meet our expectation under current conditions of hardware and dataset. However, this does not mean that there are no room for further improvement. Firstly, a common lack of large-scale dataset in MIR restrains the usage of large and advanced models recently available. More specifically, a large-scale music-lyrics dataset can be mined and created in the future, not only for the good of music genre classification, but also for other tasks like music emotion recognition. Secondly, we have already found it complementary to combine contrastive learning and MDM. Thus, it is meaningful to try different paradigms other than negative-sampling and MAE-like models. For example, BYOL may be more efficient for contrastive learning, and we may try a BEiT-like model which uses vector quantization for tokenization. Lastly, in order to learn the lyrics more effectively and better align them with audio, a bold idea is to incorporate a lyrics transcription model and the corresponding pretrained weights. In these ways, we may be able to overcome some of the existing problems and figure out a more efficient solution.

References

- [AKR15] Ritesh Ajoodha, Richard Klein, and Benjamin Rosman. "Single-labelled music genre classification using content-based features". In: *Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-ROBMECH)*. 2015, pp. 66–71.
- [Akb+21] Hassan Akbari et al. "VATT: Transformers for multimodal self-supervised learning from raw video, audio and text". In: *Advances in Neural Information Processing Systems 34 (NeurIPS)*. 2021, pp. 24206–24221.
- [Ala+20] Jean-Baptiste Alayrac et al. "Self-supervised multimodal versatile networks". In: *Advances in Neural Information Processing Systems 33 (NeurIPS)*. 2020, pp. 25–37.
- [Ara+20] Raul de Araújo Lima et al. "Brazilian lyrics-based music genre classification using a BLSTM network". In: *International Conference on Artificial Intelligence and Soft Computing*. 2020, pp. 525–534.
- [ARV20] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. "A critical analysis of self-supervision, or what we can learn from a single image". In: *International Conference on Learning Representations (ICLR)*. 2020.
- [Ash+17] Vaswani Ashish et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems 30 (NIPS)* (2017).
- [Bah18] Hareesh Bahuleyan. "Music genre classification using machine learning techniques". In: *arXiv preprint arXiv:1804.01149* (2018).
- [BAM18] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal machine learning: A survey and taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (2018), pp. 423–443.
- [BGL15] Babu Kaji Baniya, Deepak Ghimire, and Joonwhoan Lee. "Automatic music genre classification using timbral texture and rhythmic content features". In: *International Conference on Advanced Communication Technology (ICACT)*. 2015, pp. 434–443.
- [Bao+21] Hangbo Bao et al. "BEiT: Bert pre-training of image transformers". In: *International Conference on Learning Representations (ICLR)*. 2021.
- [Bao+22a] Hangbo Bao et al. "VL-BEiT: Generative vision-language pretraining". In: *arXiv preprint arXiv:2206.01127* (2022).
- [Bao+22b] Hangbo Bao et al. "VLMO: Unified vision-language pre-training with mixture-of-modality-experts". In: 2022, pp. 32897–32912.
- [BPL21] Adrien Bardes, Jean Ponce, and Yann LeCun. "VicReg: Variance-invariance-covariance regularization for self-supervised learning". In: *International Conference on Learning Representations (ICLR)*. 2021.
- [Ber+06] James Bergstra et al. "Aggregate features and ADABOOST for music classification". In: *Machine learning* 65 (2006), pp. 473–484.
- [Ber+11] Thierry Bertin-Mahieux et al. "The million song dataset". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2011, pp. 591–596.
- [Car+20] Mathilde Caron et al. "Unsupervised learning of visual features by contrasting cluster assignments". In: *Advances in Neural Information Processing Systems 33 (NeurIPS)*. 2020, pp. 9912–9924.

- [Car+21] Andrew N Carr et al. "Self-supervised learning of audio representations from permutations with differentiable ranking". In: *IEEE Signal Processing Letters* 28 (2021), pp. 708–712.
- [CJ13] Dhanith Chathuranga and Lakshman Jayaratne. "Automatic music genre classification of audio signals with machine learning approaches". In: *GSTF Journal on Computing (JoC)* 3 (2013), pp. 1–12.
- [Che+20a] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *International Conference on Machine Learning (ICML)*. 2020, pp. 1597–1607.
- [Che+20b] Ting Chen et al. "Big self-supervised models are strong semi-supervised learners". In: *Advances in Neural Information Processing Systems* 33 (*NeurIPS*). 2020, pp. 22243–22255.
- [CH21] Xinlei Chen and Kaiming He. "Exploring simple siamese representation learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 15750–15758.
- [Che+20c] Xinlei Chen et al. "Improved baselines with momentum contrastive learning". In: *arXiv preprint arXiv:2003.04297* (2020).
- [Cho+17a] Keunwoo Choi et al. "Convolutional recurrent neural networks for music classification". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 2392–2396.
- [Cho+17b] Keunwoo Choi et al. "Transfer learning for music classification and regression tasks". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2017, pp. 141–149.
- [Cir+24] Ruben Ciranni et al. "COCOLA: Coherence-Oriented Contrastive Learning of Musical Audio Representations". In: *arXiv preprint arXiv:2404.16969* (2024).
- [ÇÖ16] Önder Çoban and Gülşah Tümüklü Özyer. "Music genre classification from Turkish lyrics". In: *Signal Processing and Communication Application Conference (SIU)*. 2016, pp. 101–104.
- [con24] Wikipedia contributors. *List of music genres and styles*. https://en.wikipedia.org/w/index.php?title=List_of_music_genres_and_styles&oldid=1253415078, Last accessed on 04-11-2024. 2024.
- [COS17] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. "An evaluation of convolutional neural networks for music classification using spectrograms". In: *Applied Soft Computing* 52 (2017), pp. 28–38.
- [Den+09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255.
- [DLH23] Dorian Desblancs, Vincent Lostanlen, and Romain Hennequin. "Zero-note samba: Self-supervised beat tracking". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).
- [Dev+19] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Vol. 1. 2019, pp. 4171–4186.
- [Dos20] Alexey Dosovitskiy. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *International Conference on Learning Representations (ICLR)*. 2020.
- [Dow03] J Stephen Downie. "Music information retrieval". In: *Annual review of information science and technology* 37.1 (2003), pp. 295–340.
- [EA20] Ahmet Elbir and Nizamettin Aydin. "Music genre classification and music recommendation by using deep learning". In: *Electronics Letters* 56.12 (2020), pp. 627–629.
- [Elb+18] Ahmet Elbir et al. "Short Time Fourier Transform based music genre classification". In: *Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*. 2018, pp. 1–4.

- [Ell07] Daniel P. W. Ellis. "Classifying Music Audio with Timbral and Chroma Features". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2007, pp. 339–340.
- [Fab+07] Franco Fabbri et al. "Browsing music spaces: Categories and the musical mind". In: *Critical essays in popular musicology*. Vol. 1. 2007, pp. 49–62.
- [FS14] Michael Fell and Caroline Sporleder. "Lyrics-based analysis and classification of music". In: *International Conference on Computational Linguistics (COLING)*. 2014, pp. 620–631.
- [FRD19] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously". In: *Journal of Machine Learning Research* 20.177 (2019), pp. 1–81.
- [Fu+10] Zhouyu Fu et al. "A survey of audio-based music classification and annotation". In: *IEEE Transactions on Multimedia* 13.2 (2010), pp. 303–319.
- [Ful+18] Prasenjeet Fulzele et al. "A hybrid model for music genre classification using LSTM and SVM". In: *International Conference on Contemporary Computing (IC3)*. 2018, pp. 1–3.
- [Gan21] Jie Gan. "Music feature classification based on recurrent neural networks with channel attention mechanism". In: *Mobile Information Systems 2021* (2021), pp. 1–10.
- [GZM23] Christos Garoufis, Athanasia Zlatintsi, and Petros Maragos. "Multi-Source Contrastive Learning from Musical Audio". In: *arXiv preprint arXiv:2302.07077* (2023).
- [Gem+17] Jort F Gemmeke 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, pp. 776–780.
- [GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations". In: *International Conference on Learning Representations (ICLR)*. 2018.
- [GCG21] Yuan Gong, Yu-An Chung, and James Glass. "AST: Audio spectrogram transformer". In: *Annual Conference of the International Speech Communication Association (Interspeech)*. 2021, pp. 571–575.
- [Gon+23] Yuan Gong et al. "Contrastive audio-visual masked autoencoder". In: *International Conference on Learning Representations (ICLR)*. 2023.
- [Gri+20] Jean-Bastien Grill et al. "Bootstrap your own latent-a new approach to self-supervised learning". In: *Advances in Neural Information Processing Systems 33 (NeurIPS)*. 2020, pp. 21271–21284.
- [Har+17] Nathan Hartmann et al. "Portuguese word embeddings: Evaluating on word analogies and natural language tasks". In: *Brazilian Symposium in Information and Human Language Technology (STIL)*. 2017, pp. 122–131.
- [He+20] Kaiming He et al. "Momentum contrast for unsupervised visual representation learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 9729–9738.
- [He+22] Kaiming He et al. "Masked autoencoders are scalable vision learners". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 16000–16009.
- [Hsu+21] Wei-Ning Hsu et al. "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), pp. 3451–3460.
- [Hua+23a] Po-Yao Huang et al. "Mavil: Masked audio-video learners". In: *Advances in Neural Information Processing Systems 36 (NeurIPS)*. 2023.
- [Hua+23b] Zhicheng Huang et al. "Contrastive masked autoencoders are stronger vision learners". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

- [Kon+24] Yuexuan Kong et al. "STONE: Self-supervised Tonality Estimator". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2024.
- [KTT18] Bruno Korbar, Du Tran, and Lorenzo Torresani. "Cooperative learning of audio and video models from self-supervised synchronization". In: *Advances in Neural Information Processing Systems 31 (NeurIPS)*. 2018.
- [KRR18] Akshi Kumar, Arjun Rajpal, and Dushyant Rathore. "Genre classification using word embeddings and deep learning". In: *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2018, pp. 2142–2146.
- [LMS17] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. "Colorization as a proxy task for visual understanding". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6874–6883.
- [LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.
- [Lee+09] Chang-Hsing Lee et al. "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features". In: *IEEE Transactions on Multimedia* 11.4 (2009), pp. 670–682.
- [Li+21] Junnan Li et al. "Align before fuse: Vision and language representation learning with momentum distillation". In: *Advances in Neural Information Processing Systems 34 (NeurIPS)*. 2021, pp. 9694–9705.
- [LOL03] Tao Li, Mitsunori Ogihara, and Qi Li. "A comparative study on content-based music genre classification". In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. 2003, pp. 282–289.
- [LC11] Tom LH Li and Antoni B Chan. "Genre classification and the invariance of MFCC features to key and tempo". In: *International Conference on MultiMedia Modeling*. 2011, pp. 317–327.
- [Li+24] Yizhi Li et al. "MERT: Acoustic music understanding model with large-scale self-supervised training". In: *International Conference on Learning Representations (ICLR)*. 2024.
- [Li+23] You Li et al. "Music genre classification based on fusing audio and lyric information". In: *Multimedia Tools and Applications* 82.13 (2023), pp. 20157–20176.
- [Li+20] Yuexiang Li et al. "Self-loop uncertainty: A novel pseudo-label for semi-supervised medical image segmentation". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2020, pp. 614–623.
- [Liu+21a] Caifeng Liu et al. "Bottom-up broadcast neural network for music genre classification". In: *Multimedia Tools and Applications* 80 (2021), pp. 7313–7331.
- [Liu+23] Hong Liu et al. "M3AE: multimodal representation learning for brain tumor segmentation with missing modalities". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 2. 2023, pp. 1657–1665.
- [Liu+21b] Xiao Liu et al. "Self-supervised learning: Generative or contrastive". In: *IEEE Transactions on Knowledge and Data Engineering* 35.1 (2021), pp. 857–876.
- [LL20] Athanasios Lykartsis and Alexander Lerch. *Beat histogram features for rhythm-based musical genre classification using multiple novelty functions*. Technische Universität Berlin, 2020.
- [MNR08a] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. "Combination of audio and lyrics features for genre classification in digital audio collections". In: *International Conference on Multimedia*. 2008, pp. 159–168.
- [MNR08b] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. "Rhyme and Style Features for Musical Genre Classification by Song Lyrics." In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2008, pp. 337–342.

- [MR10] Rudolf Mayer and Andreas Rauber. "Building ensembles of audio and lyrics features to improve musical genre classification". In: *International Conference on Distributed Frameworks for Multimedia Applications*. 2010, pp. 1–6.
- [MR11] Rudolf Mayer and Andreas Rauber. "Musical genre classification by ensembles of audio and lyrics features". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2011, pp. 675–680.
- [MF06] Cory McKay and Ichiro Fujinaga. "Musical genre classification: Is it worth pursuing and how can it be improved?". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2006, pp. 101–106.
- [Meh+21] Jash Mehta et al. "Music genre classification using transfer learning on log-based mel spectrogram". In: *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. 2021, pp. 1101–1107.
- [Mik+13] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems 26 (NIPS)*. 2013, pp. 3111–3119.
- [MLN20] Pedro Morgado, Yi Li, and Nuno Nvasconcelos. "Learning representations from audio-visual spatial alignment". In: *Advances in Neural Information Processing Systems 33 (NeurIPS)*. 2020, pp. 4733–4744.
- [Nan+16] Loris Nanni et al. "Combining visual and acoustic features for music genre classification". In: *Expert Systems with Applications 45* (2016), pp. 108–117.
- [Nay22] Nikhil Nayak. *Genius Song Lyrics*. <https://www.kaggle.com/datasets/carlosgdcj/genius-song-lyrics-with-language-information>, Last accessed on 02-09-2024. 2022.
- [NF16] Mehdi Noroozi and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles". In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 69–84.
- [Ora+18] Sergio Oramas et al. "Multimodal deep learning for music genre classification". In: *Transactions of the International Society for Music Information Retrieval*. 2018 1.1 (2018), pp. 4–21.
- [Pan+21] Yagya Raj Pandeya et al. "Multi-modal, multi-task and multi-label for music genre classification and emotion regression". In: *International Conference on Information and Communication Technology Convergence (ICTC)*. 2021, pp. 1042–1045.
- [PN17] Nilesh M Patil and Milind U Nemade. "Music genre classification using MFCC, K-NN and SVM classifier". In: *International Journal of Computer Engineering In Research Trends 4.2* (2017), pp. 43–47.
- [PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543.
- [Qui22] Elio Quinton. "Equivariant self-supervision for musical tempo estimation". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2022, pp. 84–92.
- [Rad+21] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning (ICML)*. 2021, pp. 8748–8763.
- [Rag+21] Maithra Raghu et al. "Do vision transformers see like convolutional neural networks?". In: *Advances in Neural Information Processing Systems 34 (NeurIPS)*. 2021, pp. 12116–12128.
- [Ren+16] Shaoqing Ren et al. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence 39.6* (2016), pp. 1137–1149.

- [Rio+23] Alain Riou et al. "Pesto: Pitch estimation with self-supervised transposition-equivariant objective". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2023, pp. 535–544.
- [Ru+23] Ganghui Ru et al. "Improving music genre classification from multi-modal properties of music and genre correlations perspective". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5.
- [San+20] Igor André Pegoraro Santana et al. "Music4all: A new music database and its applications". In: *International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE. 2020, pp. 399–404.
- [SS15] Rajib Sarkar and Sanjoy Kumar Saha. "Music genre classification using EMD and pitch based feature". In: *International Conference on Advances in Pattern Recognition (ICAPR)*. 2015, pp. 1–6.
- [SGU+14] Markus Schedl, Emilia Gómez, Julián Urbano, et al. "Music information retrieval: Recent developments and applications". In: *Foundations and Trends® in Information Retrieval* 8.2-3 (2014), pp. 127–261.
- [SR15] Alexander Schindler and Andreas Rauber. "An audio-visual approach to music genre classification through affective color features". In: *Advances in Information Retrieval: 37th European Conference on IR Research (ECIR)*. 2015, pp. 61–67.
- [SLT19] Leisi Shi, Chen Li, and Lihua Tian. "Music genre classification based on chroma features and deep learning". In: *International Conference on Intelligent Control and Information Processing (ICICIP)*. 2019, pp. 81–86.
- [SD14] Siddharth Sigtia and Simon Dixon. "Improved music feature learning with deep neural networks". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 6959–6963.
- [Sin+22] Amanpreet Singh et al. "FLAVA: A foundational language and vision alignment model". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 15638–15650.
- [SB21] Janne Spijkervet and John Ashley Burgoyne. "Contrastive learning of musical representations". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2021, pp. 673–681.
- [Sun+19] Chen Sun et al. "VideoBERT: A joint model for video and language representation learning". In: *International Conference on Computer Vision (ICCV)*. 2019, pp. 7464–7473.
- [Sze+17] Christian Szegedy et al. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.
- [Tao+23] Chenxin Tao et al. "Siamese image modeling for self-supervised vision representation learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 2132–2141.
- [Tay53] Wilson L Taylor. "'Cloze procedure': A new tool for measuring readability". In: *Journalism quarterly* 30.4 (1953), pp. 415–433.
- [TLR23] Bernardo Torres, Stefan Lattner, and Gael Richard. "Singer Identity Representation Learning using Self-Supervised Techniques". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2023, pp. 448–456.
- [Tou+21] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning (ICML)*. 2021, pp. 10347–10357.
- [Tsa17] Alexandros Tsaptsinos. "Lyrics-based music genre classification using a hierarchical attention network". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2017, pp. 694–701.
- [TC02] George Tzanetakis and Perry Cook. "Musical genre classification of audio signals". In: *IEEE Transactions on Speech and Audio Processing* 10.5 (2002), pp. 293–302.

- [TEC03] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. "Pitch histograms in audio and symbolic music information retrieval". In: *Journal of New Music Research* 32.2 (2003), pp. 143–152.
- [VM22] Igor Vatolkin and Cory McKay. "Multi-Objective Investigation of Six Feature Source Types for Multi-Modal Music Classification." In: *Transactions of the International Society for Music Information Retrieval* 5.1 (2022), pp. 1–20.
- [WM21] Laisha Wadhwa and Prerana Mukherjee. "Music genre classification using multi-modal deep learning based fusion". In: *Grace Hopper Celebration India (GHCI)*. 2021, pp. 1–5.
- [Wan+09] Fei Wang et al. "Tag Integrated Multi-Label Music Style Classification with Hypergraph". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2009, pp. 363–368.
- [Wan+22a] Luya Wang et al. "Repre: Improving self-supervised vision transformer with reconstructive pre-training". In: *International Joint Conference on Artificial Intelligence (IJCAI)*. 2022, pp. 1437–1443.
- [Wan+22b] Wenhui Wang et al. "Image as a foreign language: Beit pretraining for all vision and vision-language tasks". In: *arXiv preprint arXiv:2208.10442* (2022).
- [Wu+21] Ho-Hsiang Wu et al. "Multi-task self-supervised pre-training for music classification". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 556–560.
- [Xie+23] Zhenda Xie et al. "On data scaling in masked image modeling". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 10365–10374.
- [YWG22] Hiromu Yakura, Kento Watanabe, and Masataka Goto. "Self-supervised contrastive learning for singing voices". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2022), pp. 1614–1623.
- [Yan+20] Rui Yang et al. "Parallel recurrent convolutional neural networks-based music genre classification method for mobile devices". In: *IEEE Access* 8 (2020), pp. 19629–19637.
- [Yu+20] Yang Yu et al. "Deep attention based music genre classification". In: *Neurocomputing* 372 (2020), pp. 84–91.
- [Zbo+21] Jure Zbontar et al. "Barlow twins: Self-supervised learning via redundancy reduction". In: *International Conference on Machine Learning (ICML)*. 2021, pp. 12310–12320.
- [Zel+22] Rowan Zellers et al. "Merlot reserve: Neural script knowledge through vision and language and sound". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 16375–16387.
- [Zha21] Kedong Zhang. "Music style classification algorithm based on music feature extraction and deep neural network". In: *Wireless Communications and Mobile Computing* 2021 (2021), pp. 1–7.
- [Zha+16] Weibin Zhang et al. "Improved music genre classification with convolutional neural networks." In: *Annual Conference of the International Speech Communication Association (Interspeech)*. 2016, pp. 3304–3308.
- [Zha+22] Hang Zhao et al. "S3t: Self-supervised pre-training with swin transformer for music classification". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 606–610.
- [ZMM17] Eve Zheng, Melody Moh, and Teng-Sheng Moh. "Music genre classification: A n-gram based musicological approach". In: *International Advance Computing Conference (IACC)*. 2017, pp. 671–677.
- [Zho+21] Jinghao Zhou et al. "iBOT: Image bert pre-training with online tokenizer". In: *arXiv preprint arXiv:2111.07832* (2021).
- [Zhu+21] Hongyuan Zhu et al. "MusicBERT: A self-supervised learning of music representation". In: *International Conference on Multimedia*. 2021, pp. 3955–3963.

- [ZCZ20] Yingying Zhuang, Yuezhang Chen, and Jie Zheng. “Music genre classification with transformer classifier”. In: *International Conference on Digital Signal Processing*. 2020, pp. 155–159.
- [ZMH23] Yongshuo Zong, Oisin Mac Aodha, and Timothy Hospedales. “Self-supervised multimodal learning: A survey”. In: *arXiv preprint arXiv:2304.01008* (2023).