AN ABSTRACT OF THE THESIS OF

Shaun K. Leib for the degree of Master of Science in Artificial Intelligence presented on April 28, 2026.

Title: Time is Not on Your Side: Assessing the Value of Dynamic Audio Features for Static Music Emotion Labels

Abstract approved: _____

Patrick Donnelly

Music Emotion Recognition is an unsolved task within the field of Music Information Retrieval with a relatively stagnating state of the art. Within the problem there are two sub-tasks which we explore. Static emotion labeling considers a global predominant emotion label for the entire audio, whereas dynamic emotion labeling, or music emotion variance detection, considers emotion fluctuations over time. In our work, we analyze the embeddings of three pre-trained audio models (Wav2Vec 2.0, MERT, MAEST) in addition to handcrafted low/mid-level features to determine if temporally dynamic features can lead to higher performance on static emotion labels. Specifically, we examine if the information gained from dynamic features over temporally averaged global features from the same models lead to increased performance on statically labeled music. These experiments are performed utilizing the "Music Emotion Recognition next GEneration (MERGE)"

dataset, a recently created music dataset which removes many of the issues prevalent in previous MER datasets. In addition, we test three levels of feature fusion in order to examine how our feature types interact, and we obtain the highest audio only performance on the MERGE dataset. Counter to our beliefs, we find that temporally dynamic features do not add additional information for static emotion labels.

©Copyright by Shaun K. Leib
April 28, 2026
All Rights Reserved

Time is Not on Your Side: Assessing the Value of Dynamic Audio Features for Static Music Emotion Labels

by

Shaun K. Leib

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented April 28, 2026
Commencement June 2026

Master of Science thesis of Shaun K. Leib presented on April 28, 2026.

APPROVED:

Major Professor, representing Artificial Intelligence

Head of the School of Electrical Engineering and Computer Science

---

Dean of Graduate Education

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Shaun K. Leib, Author

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor Patrick Donnelly, for assisting me throughout this entire project. From start to finish, without his help this would not have been possible. Thank you mom and dad for your continued support. I know I can always count on you both. I would also like to thank all the members of our Soundbendor Lab for helping me polish my work and presentations. Finally, thank you to my committee for taking time out of their busy schedules to read and listen to my work.

TABLE OF CONTENTS

Page

1 Introduction 1
1.1 Automatic Emotion Classification 2
1.2 Our Efforts 4

2 Background 6
2.1 Recurrent Neural Networks 6
2.2 Convolutional Neural Networks 9
2.3 Attention 11
2.4 Models of Emotion 12

3 Related Works 16
3.1 MediaEval 16
3.2 Feature Fusion 19

4 Dataset 24

5 Methods 28
5.1 Handcrafted Features 28
5.2 Pre-Trained Models 34
5.2.1 Wav2Vec 2.0 34
5.2.2 MERT 36
5.2.3 MAEST 39

6 Experiments 41
6.1 Handcrafted Feature Optimization 42
6.2 Dynamic vs. Static 45
6.2.1 Dynamic Results 46
6.2.2 Static Features 50
6.3 Fusion 52
6.4 Regression 58

TABLE OF CONTENTS (Continued)

Page

7 Discussion 61
7.1 Dynamic vs. Static 61
7.2 Fusion 65
7.3 Regression 66
7.4 Limitations 67
7.5 Future Work 69

8 Conclusion 70

Bibliography 72

LIST OF FIGURES

Figure | Page

2.1 A visualization of a LSTM cell, recreated from [56]. This figure originally appears in our previous work [29]. 8

2.2 A visualization of a single convolution, recreated from [23]. In the example highlighted in red, 5.3 = 1.2 + 0 + 2.1 + 0 + 1.0 + 0 + 0 + 0 + 1.0. The exact kernel values are learned during training. This figure originally appears in our previous work [29]. 9

2.3 Russell's graph of Valence and Arousal [45]. This figure originally appears in our previous work [29]. 15

5.1 A visualization of the Wav2Vec 2.0 framework which learns both contextualized speech representations and quantized speech units. Figure from [3]. 36

5.2 A visualization of the MERT pre-training framework which utilizes a student teacher framework with acoustic and musical teachers. During pre-training, the model attempts to reconstruct masked time steps to maximize teacher clustering and MSE. Figure from [32]. . . 37

5.3 A visualization of the supervised training and downstream evaluation procedure of MAEST. Trainable blocks are shown in green and frozen blocks in blue. Figure from [2]. 38

6.1 A visualization of the mutual information for each feature over time. 43

6.2 Our standard dynamic model. The Encoder layers inside the denoted dotted line are reused in further figures. 47

6.3 The model utilized for the MAEST embeddings. Only the sequence tokens are input into the encoder block and concatenated with the classification and distillation tokens before pooling. 47

6.4 This confusion matrix shows the distribution of our classification results when utilizing the dynamic MAEST embeddings. 50

6.5 This confusion matrix shows the distribution of results from our static MAEST embeddings, or our best results. 52

6.6 Our late fusion architecture which combines the four separate inputs. 54

LIST OF FIGURES (Continued)

Figure		Page
6.7	Our mid-level fusion model which utilizes cross attention to fuse inputs.	55
6.8	This confusion matrix shows the distribution of class predictions for our static late fusion results.	57
7.1	The distribution of our regression predictions when utilizing static MAEST embeddings.	67

LIST OF TABLES

Table		Page
2.1	Three common psychological models of emotion used in MER.	13
4.1	The distribution of audio samples between the different versions of the MERGE dataset. Recreated from [34].	25
4.2	Baseline performance metrics for the MERGE dataset provided by [34] [33] [35].	26
5.1	The distribution of our handcrafted features by musical dimension.	29
6.1	Results using different feature selection and dimensionality reduction techniques for our handcrafted set of features.	44
6.2	Results of our experiments using dynamic features extracted from handcrafted methods and three pre-trained models.	48
6.3	Results from downsampling our MAEST embeddings to 60 and 30 time steps.	49
6.4	Results from employing other dynamic models with MAEST embeddings.	49
6.5	Results from our static features can be seen in the top half of the table. The handcrafted features are the same set from [29], and the features from pre-trained models are obtained by averaging their temporal dimension. The scores on the bottom half reflect the baseline metrics for the MERGE dataset, also obtained with static representations.	51
6.6	Results from different fusion strategies on our dynamic features.	55
6.7	Results of different fusion strategies on our static features.	56
6.8	Ablation results from late fusion of different subsets of static feature inputs.	58
6.9	Results from our comparison of dynamic and static MAEST embedding on continuous regression targets. Our audio only static results represent the state-of-the-art performance on MERGE, outperforming the bimodal baseline.	59

LIST OF TABLES (Continued)

Table Page

6.10 The standard deviation of our static regression results from all 20 independent trails. 60

Chapter 1: Introduction

Music Emotion Recognition, or MER, is a subfield within the Music Information Retrieval field in which computational methods attempt to identify and predict common emotional responses evoked from music. The field combines several disciplines, including music theory, psychology, signal processing, and artificial intelligence with each contributing insight into how music and sound are connected to emotions. This is done in an attempt to bridge the gap between human emotional experiences and machine understanding of musical content, as well as to improve upon manual human annotations of songs.

Traditionally, manual emotional annotations have been obtained through careful human subject studies in a controlled research lab or from crowd-sourcing platforms like Amazon Mechanical Turk. However, these methods are often expensive and time-consuming. Furthermore, they often do not reliably produce consistent results across independent annotations, as different annotators interpret the emotional content of the same piece of music differently. These problems can be seen in early and widely used MER datasets like DEAM [1] and AMG1608 [10]. These datasets were limited by manual annotation methods, which resulted in a small amount of samples and a considerable degree of disagreement between annotators. Despite these drawbacks, such datasets remain valuable and have historically contributed to MER research. However, their constraints and limitations highlight the

need for automatic annotation approaches that can scale to larger music datasets.

1.1 Automatic Emotion Classification

Automatic emotion classification has a wide range of real-world applications across many domains. For example, one of the most important applications is the implementation of emotion-based music recommendation systems and mood-based playlist generation [13]. Other areas include affective music therapy applications [18] and systems for generative music composition in which the goal is to generate a piece of music that evokes a target emotion [12]. As a result of these areas, MER has become an important area of research within Music Information Retrieval, or MIR, with the goal of bridging the gap between subjective human experience and machine understanding of emotion.

To address this, many studies have explored different approaches to computationally classifying emotion in music. The first consideration is how to represent the audio for input into a model. Raw audio waveforms by themselves are highly complex and high-dimensional. This makes it difficult for models to directly identify any emotional patterns from all the noise within them. As a result, many past studies have utilized alternative representations that capture structured patterns within the music. These include acoustical features with music theory backing [41] [36] [54] [11] [52], lyrical content [53], spectrogram image visualizations [44] [11], and combinations of multiple modalities [13]. Recently, with the advent of Large Audio Language Models, which integrate the successful principals learned from

LLMs into the audio space, raw audio and spectrogram embeddings have proven to be useful for many MIR tasks [3] [32] [2].

Different input representations are not the only area in which advancements have been made. In addition, a variety of machine and deep learning models have been applied to classify these features. Many early studies frequently utilized conventional ML models like support vector machines (SVMs) [41], however, more recent work has adopted deep learning architectures which are capable of modeling more complex temporal and spectral patterns [30] [14] [31].

Finally, many different dataset representations have been explored as well. This includes how songs are labeled according to different emotional taxonomies. These taxonomies range from those with psychological backing to custom multi-tag classification or regression.

Despite these efforts, progress within many MIR subfield including MER have experienced a stagnation. Scholz, Ramalho, and Giordano argue this stagnation stems from both a lack of musical knowledge as well as a slow adoption of more advanced deep learning techniques [46]. In a similar manner, Panda et al. [41] argue that many of the difficulties in MER specifically arise from the inherent ambiguity in human emotional perception and the absence of a widely recognized public dataset with a psychological taxonomy. With all of this in mind, these challenges further motivate the research presented here into more effective audio representations and models for automatic music emotion classification.

1.2 Our Efforts

Our work builds upon our previous effort [29] in which we extracted a large set of handcrafted features to classify emotion in the 4QAED [41] dataset, a predecessor to the one used here. The feature set used in our previous study was global static features obtained by statistically aggregating sequential dynamic acoustic features. We utilized Recursive Feature Elimination to reduce our feature set to the top 300 features to obtain an F_1 score of 83.03%, the state-of-the-art score on the dataset.

In this study, we present a comparison between static global-level features versus dynamic local-level features from a variety of different sources including handcrafted acoustic features, raw audio embeddings, and spectrogram embeddings. We utilize the "Music Emotion Recognition next GEneration (MERGE)" dataset from Louro et al. [34] and compare our results to their baselines. To generate our results, we employ a variety of different machine and deep learning models including MLPs, CNNs, LSTMs, and Transformers. In addition to these, we develop several fusion models (Early, Mid, Late) which utilize several inputs to formulate a single prediction. These models help us form a conclusion contrary to our initial hypothesis: static features slightly outperform dynamic ones on global level emotion prediction. Despite this, we still achieve state of the art audio only performance on the MERGE dataset with both static and dynamic features, outperforming all baselines. In addition to this classification task, we also demonstrate the ability to correctly predict regression values. This highlights a potential limitation of the MERGE dataset, namely that the task may be comparatively "easy."

The MERGE dataset is bimodal, containing both raw audio as well as transcribed lyrics, however, we restrict our experiments to audio only. While lyrical content certainly contributes to the emotion of a song, this is beyond the scope of this study. Within MER, acoustic and lyric features are considered two separate domains which are both being researched. Most importantly, lyrics are not temporally dynamic in the same manner as acoustic features. As we are specifically studying the effects of dynamic features versus static ones, we do not include them.

Overall, this study will start by describing background information in Section 2. This provides a detailed summary of the models utilized in this study as well as different psychological models of emotion. The next Section 3, related works, details more background information required to understand the decisions we made in our process. In this, we analyze prior work on past studies in MER that have led up to our experiments, including the DEAM dataset [1] and different feature fusion results. Next, Section 4 describes the MERGE dataset used for all our experiments. We provide a background for why the dataset was created in the first place, and how it improves upon past datasets. Section 5 explains how we extract and process both our handcrafted feature set and the embeddings from the three pre-trained models utilized in our experiments. These features and embeddings are then used in the experiments described in Section 6. We begin this section by describing the metrics used to evaluate our results. This is followed by feature selection experiments for our handcrafted feature set, and the main dynamic versus static experiments. Finally, Section 7 discusses the implications of our results, our contributions, the limitations of the study, and directions for future work.

Chapter 2: Background

In this chapter, we provide preliminary background information on many of the topics required to complete this work. We describe the deep learning methods used in our experiments, as well as different models of emotion.

2.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of neural network specially designed to capture temporal, or sequential information, by introducing explicit dependencies between units. This is accomplished through self-referencing cycles and parameter sharing across time, which allows previous steps to influence the current one. Formally, each hidden state is updated by utilizing both the current input and previous hidden state [56]:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b)$$

where x_t denotes the current input at time t, h_t is its corresponding hidden state, and W_h, W_x, and b are learnable parameters.

RNN architectures are typically categorized by the input they receive and the output they compute. *One-to-many* models take a single input (e.g., an image) and output a sequence (e.g., an image caption). *Many-to-one* models do the opposite,

processing a sequence of inputs to compute a single output, and are often used in classification or forecasting tasks. Finally, the *Many-to-many* configuration takes sequences as inputs and produces sequences as outputs, as is common in text processing tasks. Regardless of the types of inputs and outputs, training these models requires a variant of the standard backpropagation known as backpropagation-through-time, which accounts for the recurrent connections in the layers.

However, RNNs are susceptible to vanishing and exploding gradients which can cause the training of larger models to underperform. This is due to the gradient values becoming extremely small or large as it is backpropagated through the network. As a consequence, longer input sequences in standard RNNs lose their ability to capture long-range dependencies. To address this problem, Long Short-Term Memory (LSTM) networks were created as a specialized variation of RNN. They introduce internal state variables to their nodes which affect their output. This state, commonly referred to as a cell, allows the network to retain more information across longer and more complex inputs.

Within each cell, there are three main components: the `forget gate`, the `input gate`, and the `output gate`, as shown in Figure 2.1. As its name suggests, the forget gate determines the information deemed contextually important that should be retained in the cell and determines which information can be forgotten. The input gate controls what new information in the sequence should be allowed into the state, while the output gate decides what information in the current state should output for the current timestep. Each of these gates takes on a value between 0 and 1, where 0 represents the complete blockage of information, and

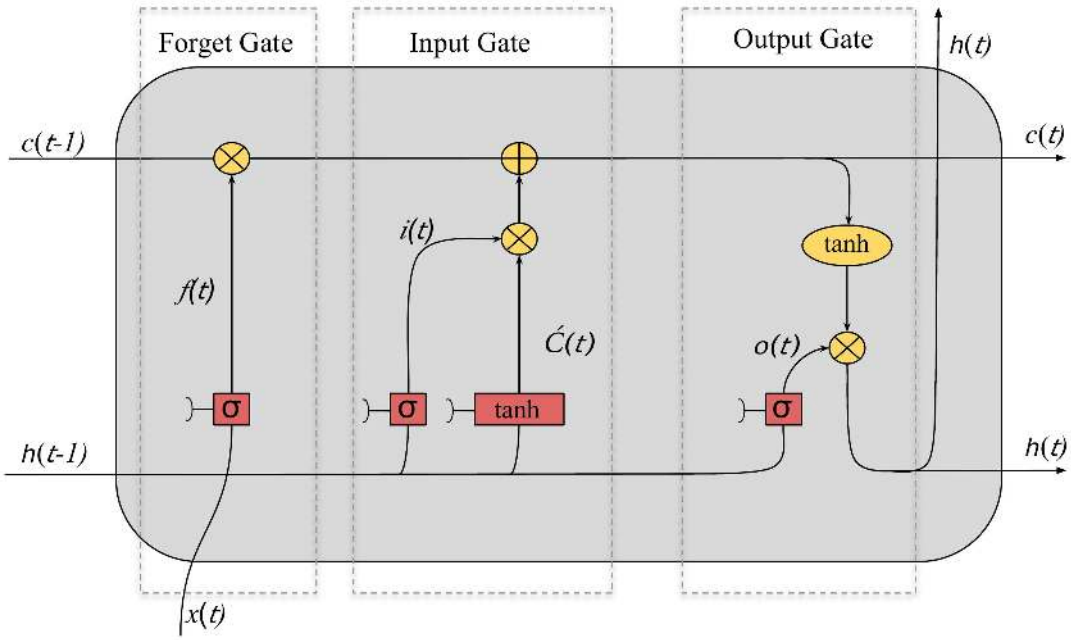


Figure 2.1: A visualization of a LSTM cell, recreated from [56]. This figure originally appears in our previous work [29].

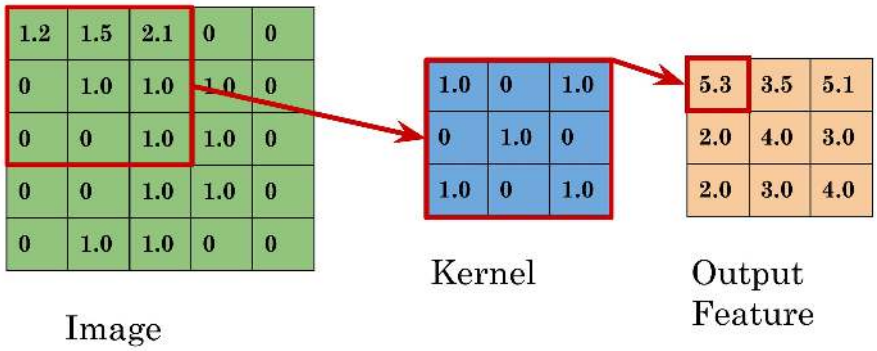


Figure 2.2: A visualization of a single convolution, recreated from [23]. In the example highlighted in red, $5.3 = 1.2 + 0 + 2.1 + 0 + 1.0 + 0 + 0 + 0 + 1.0$. The exact kernel values are learned during training. This figure originally appears in our previous work [29].

1 allows all information to pass. Through this information regulation, LSTMs significantly increase an RNN's ability to "remember" information across the entire sequence, and help mitigate the gradient-related issues [56].

Finally, RNNs, regardless of type, can operate bidirectionally. This option allows sequences to be processed both forward and backward by maintaining two independent states that are trained simultaneously. This extension allows RNN's to incorporate both past and future contextual information from the sequence, often leading to better results [47].

2.2 Convolutional Neural Networks

Another type of artificial neural network discussed in this work is the convolutional neural network (CNN). Originally popularized by their success in computer vision tasks, these networks are unique due to the convolution operation they employ.

Originally an operation from signal processing, a convolution consists of a learnable kernel or filter function that is applied to an input in order to extract meaningful features. These kernels are typically smaller in size than the input and operate as a sliding window systematically stepping over the input. As this kernel moves across the input, the dot product between the kernel weights and the corresponding input space is calculated at each position and stored as the output, as shown in Figure 2.2. Because a CNN layer still computes a linear combination of learnable weights and the input, another way to think of a CNN layer is as a specialized fully connected layer where only a specific pattern of weights is learned, and the rest are always 0.

Because this operation can downsample the output, there are many options to further down-sample or keep the output the same size. These include the stride, padding, and size of the kernel which are all hyperparameters set before training. The stride hyperparameter determines how much the filter shifts between operations. Higher stride values lead to more heavily downsampled outputs. Padding, on the other hand, adds rows and columns of zeros around the input in order to keep the dimensionality the same or even upscale the output. Finally, the kernel size itself affects the output dimensionality. In most modern practice, smaller filter sizes (e.g., 3 x 3) are used for earlier layers, and may be larger in later layers. This is because in computer vision and signal processing, convolutions are used to extract the relevant features and patterns in the input. Earlier layers with smaller kernels have smaller receptive fields and extract lower-level features (e.g., edge detection). In deeper levels, receptive fields are much larger and as such,

their outputs capture much higher level features. Finally, pooling layers, such as mean or max pooling, are often added following convolutional layers, further reducing dimensionality [57]. The extracted features from CNNs are then passed as input to other types of neural networks like MLPs and RNNs for classification and regression.

2.3 Attention

Attention is a mechanism that allows a model to selectively focus on the most relevant information. This is accomplished by computing weights that reflect the importance of each part of the input sequence. Mathematically, this is done using three separate learned matrices. For a given token or time step, the Query matrix represents the information that the token is searching for in the rest of the sequence. The Key matrix, on the other hand, represents the available information of other tokens in the sequence, and the Value matrix contains the actual content. These matrices are obtained from learned projection matrices and are used in conjunction in the following attention function:

$$A = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

where d_k is used to prevent large values from the inner product. The softmax function is essential to produce normalized attention weights from similarity scores, allowing multiple tokens to attend to different parts of the sequence.

In addition to the traditional self-attention mechanism, our experiments utilize temporal attention pooling. With this approach, a scalar importance weight is calculated for each time step to produce a weighted summary of the sequence. Given a sequence $H = [h_1, h_2, ..., h_T], h_t \in \mathbb{R}^d$, where T is the number of time steps and d is the feature dimension, the attention scores are computed as:

$$e = \tanh(HW + \mathbf{1}b^T)u$$

in which W, b, and u are learnable parameters. In a manner similar to dot-product attention, the softmax function is used to obtain weights that are multiplied by the original sequence values.

$$c = H^T \ \text{softmax}(e)$$

This pools the temporal dimension into a single feature vector, c, based on learned weights, thereby determining which time steps in the sequence matter most for the final prediction [4].

2.4 Models of Emotion

In this section, we provide an overview of the development of different emotion representation models used within the music emotion recognition (MER) community. These models originate in psychology and have been adapted for their ability to discretize or quantify different emotions into separate categories or dimensions.

These categories and dimensions are argued by their creators to represent fundamental emotions which can be used to describe all others. Within MER, three commonly used models are summarized in Table 2.1.

Author	**Year**	**Type**	**Emotions**
Russell [45]	1980	Continuous	Valence, Arousal, Dominance
Plutchik [43]	1980	Categorical	Fear, Anger, Joy, Sadness, Acceptance, Disgust, Expectation, Surprise
Ekman [15]	1992	Categorical	Fear, Anger, Joy, Sadness, Disgust, Surprise

Table 2.1: Three common psychological models of emotion used in MER.

Plutchik's and Ekman's models rely on categorical representations of emotion that they argue are the most fundamental aspects of human emotional experience. In Plutchik's book *Theories of Emotion*, he defines these fundamental emotions from an evolutionary perspective as those that conferred adaptive advantages. He determined that there are four opposing pairs of basic emotions which are Fear-Anger, Joy-Sadness, Acceptance-Disgust, and Expectation-Surprise. He arrived at these eight emotions by arguing that a basic emotion should be one which is triggered immediately by external stimuli. Importantly, his emotions are not synonymous with internal states of feeling, but the automatic response mechanisms activated as a result of the environment [43]. In addition to the four pairs of fundamental emotions, Plutchik describes the behaviors, actions and personality traits associated with them. For example, an external stimulus such as the threat

of a predator would trigger the cognitive recognition of danger, leading to the feeling of *fear*. This immediate emotional response drives behavior and influences an organism's actions. In this example, the associated behavior of *fear* is *to run*, and the trait of an organism prone to *fear* is *skittish* [43].

Later developed in 1992, Ekman's emotional model critiques Plutchik's model while still using the same framework of fundamental emotions. Specifically, Ekman's model excludes the acceptance and expectation pair, while keeping the other six emotions. He argues that acceptance and expectation do not meet the criteria of fundamental emotions because they are not an immediate and distinct set of neural signals that are distinguishable from long-term traits and personalities. For Ekman, a basic emotion must focus solely on current external stimuli, and as such, acceptance and expectation should not be considered as they involve future or past actions. Ekman then argues that Fear, Anger, Joy, Sadness, Disgust, and Surprise satisfy nine characteristics: Distinctive universal signals, Presence in other primates, Distinctive physiology, Distinctive universals in antecedent events, Coherence among emotional responses, Quick onset, Brief duration, Automatic appraisal, Unbidden occurrence [15]. Finally, Ekman states that these six emotions encompass a family of related emotions which share similar characteristics, however, he does not address any mapping between these groups [15].

The final model differs from the others in the sense that it does not use discrete classes to define basic emotions. Instead, emotions are represented using continuous values across three independent dimensions: Valence, Arousal, and Dominance, as illustrated in Figure 2.3. First proposed in 1957 by Osgood, this model argues

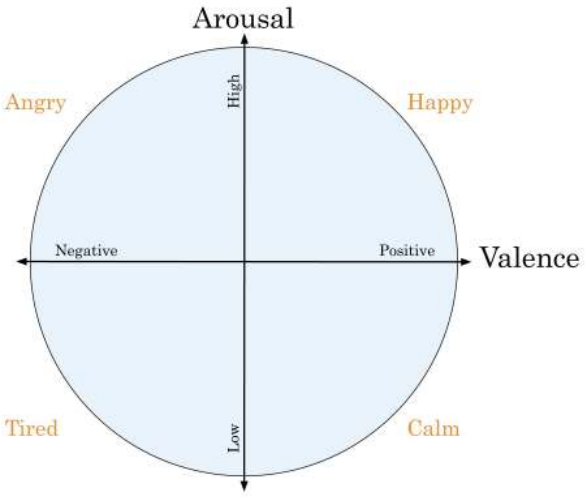


Figure 2.3: Russell's graph of Valence and Arousal [45]. This figure originally appears in our previous work [29].

that these three dimensions could describe any emotion though a combination of their axes [40]. Later in 1980, James Russell expanded upon this idea by arranging the three dimensions perpendicularly within a unified emotion space known as the Circumplex Model of Emotion. His work demonstrates how the dimensions are not entirely independent from one another, and how previous categorical models, such as Plutchik's and Ekman's, can be effectively represented within his model. Importantly, the emotions described in Plutchik's and Ekman's models require only the Valence and Arousal dimensions [45]. In later works, the third dimension, Dominance, has been shown to be strongly correlated with the Valence dimension [50], and as a result, many studies utilizing this model, including our own, omit it.

Chapter 3: Related Works

This chapter reviews previous MER studies published prior to our research. It provides an in-depth review of different machine learning and deep learning models, input modalities, and datasets.

3.1 MediaEval

To understand the work that has been undertaken in Music Emotion Recognition, it is important to understand the "Emotion in Music" task in the MediaEval Multimedia Evaluation Campaign from 2013 to 2015. During this period, researchers created the Database for Emotional Analysis in Music (DEAM), one of the first widely used datasets, and organized a competition to identify the best-performing solutions [1]. The DEAM dataset consists of 1,744 45-second songs excerpts with some full songs being added a couple of years later. The dataset was labeled with Russell's Valence-Arousal model using continuous values for both dimensions, with both static and dynamic labels for each song. The dynamic labels begin after 15 seconds and continue until the end of the song with labels assigned every half second. These labels were obtained using Amazon's Mechanical Turk services, and the dataset is imbalanced and lacks quality control for annotator agreement. In addition to the raw audio and labels, the dataset also provides a set of 260 low-level

features extracted from OpenSMILE. Despite numerous criticisms of the dataset, the competition led to many early contributions and set the stage for future work. Notably, the results from the early competitions [1] found that recurrent networks such as LSTMs are particularly effective in modeling the dynamic changes in emotion from the OpenSMILE acoustic features. Additionally, researchers found that modeling arousal can be done effectively using simple timbrel features but the valence dimension is much more complex. In the final 2015 competition, studies [31] [30] highlighted the importance of using a bidirectional LSTM for both future and past contextual representations of the songs, and different late-stage fusion strategies with extreme learning machines, support vector machines, or multi-layer perceptrons could further increase results. Specifically, Li et al. [30] found that using dynamic features to predict dynamic labels can be done more effectively by post-processing the contextual BLSTM information with triangular filter smoothing, and modeling hierarchical structure by fusing different LSTM models operating at different time scales into a single prediction. These efforts led to the best results on the dataset in the 2015 competition with an RMSE of 0.285 for valence and 0.225 for arousal.

Even after the competition ended, studies continued to utilize the dataset. Pons et al. [44] demonstrated that CNNs can extract relevant temporal and frequency information from spectrograms. Using this information, several teams found that stacked CNN-RNN models further increased performance. Malik et al. [38] developed a CNN GRU model which outperformed Li et al. [30]. Later, Yizhuo Dong et al. utilized BLSTMs again instead of GRUs to achieve comparable performance

using affect-salient features [14]. Finally, [37] argued that there is no one-to-one mapping between music and valence or arousal in the time domain as the emotion of a song at one point is influenced by the music before it. They investigated how much preceding context provides useful information for emotion prediction. In order to test this, they compared a multi-scale context based attention model with a single-scale context based attention model for static emotion prediction. Their single-scale model is essentially a generic LSTM model with attention pooling to obtain a single prediction, in which each hidden LSTM layer, h_t, has a feature input x_t. However, their multi-scale model combines many single-scale LSTM models with each LSTM receiving input sequences of varying lengths (i.e. $LSTM_1$ has $x_1, ..., x_t$ and $LSTM_2$ has $x_2, ..., x_t$). In this manner, the model itself determines how much previous context to include, without having to manually pool or average time steps together. They utilized the provided OpenSMILE features and found that the multi-scale model outperformed the single-scale model by 0.003 RMSE for valence and 0.031 RMSE for arousal. Finally, they found that the attention pooling mechanism to determine which time steps are most important before classification significantly helped.

Years later, the DEAM dataset is still used, and newer architectures and strategies continue to be tested with this older dataset. In 2024, Chen et al. used a transformer encoder architecture with positional, token, and temporal encodings to extract both local and global context representations from the provided OpenSMILE feature set [9]. They also found that a self-attention pooling layer which combines the dynamic features into a static set to be beneficial for static emotion

labeling. While the arousal RMSE decreased to 0.2236, their model was able to reduce the valence RMSE to 0.2466 for the static labels. For dynamic labeling, in 2025 Jiang and Zhang used two different inputs, Mel Spectrograms, which have been widely used before for acoustic features, and Cochleograms, a different visual input that represents physiological texture cues [20]. They used separate CNN architectures to first process each input independently, then fused these two representations together as input to a transformer. This is the most effective model for dynamic music emotion recognition for the DEAM dataset with 0.168 and 0.139 RMSE for valence and arousal respectively.

3.2 Feature Fusion

Many other advancements in MER have taken place in areas outside the DEAM dataset. Other datasets include the MERGE dataset used in this study, the multi-class MTG Jamendo Mood sub-dataset [7] and MagnaTagATune Mood sub-dataset [26], as well as a multitude of other datasets that provide different input types (e.g., music videos) and labeling taxonomies. Additionally, like many other tasks, many state-of-the-art models in MER are now leveraging multi-modal feature fusion in order to increase their performance, although not all of the fusion strategies and modality combinations are equally effective.

The dataset used in our experiments is the "Music Emotion Recognition next GEneration (MERGE)" dataset, which is further described in Section 4. MERGE was created to solve many of the issues inherent in the pre-existing datasets. These

problems include low sample sizes, no psychologically grounded emotional taxonomy, labeling methods without human validation or quality control, lacking strong and clear baselines, and missing the raw audio itself. This dataset builds upon their previous 4QAED [41] dataset which contained only 900 samples. MERGE saw the addition of many more samples, bringing the total up to 3,554 songs with raw audio and lyrics, statically labeled according to the four quadrants of Russell's Circumplex Model of Emotion [34]. Four versions of the dataset are provided, depending on the task: two multimodal versions combining audio and lyrics and two larger audio-only versions. In addition, there is a complete version and a balanced version, with the latter constituting a subset of the former and having an equal label distribution. For the audio only dataset, the initial baseline provided in the original paper [34] for quadrant classification is a 70.18% F_1-score. For regression tasks, they provide baselines of 0.479, 0.364 R^2; and 0.281, 0.375 RMSE, for arousal and valence respectively. These were obtained using 1704 static features reduced with ReliefF feature selection and a SVM. For the bimodal version, they found that lyric features and lyric embeddings significantly help with the valence dimension. With audio embeddings from the Wav2Vec2.0 model combined with lyric embeddings from RoBERTa, they were able to achieve 81.74% F_1 for quadrant classification, and 0.505, 0.528 R^2; and 0.169, 0.305 RMSE for arousal and valence, respectively, on the regression prediction.

With these baselines, further work was published utilizing the fusion of different input methods. In [33], they combined the same handcrafted features with learned features extracted from mel spectrograms using a CNN model. These two

feature types were then fused before classification, achieving a 77.62% F_1-score. In another study [35], the authors extracted audio embeddings from the Wav2Vec 2.0 and MERT models, and lyric embeddings from the RoBERTa and ModernBERT models. They experimented with both early and late feature fusion, and found that late-stage majority-voting fusion resulted in their best results of 79.21% weighted F_1.

Other types of fusion have borrowed from the closely related task of speech emotion recognition (SER). Many of these studies have utilized mid-level cross-feature fusion with attention. For example, Li and Zhang [28] made use of an attention masked residual network (AM-ResNet) to filter out silent and unvoiced audio frames and combine them with Wav2Vec 2.0 embeddings using cross attention. The key, query, and value matrices were obtained for each feature type using learned projection matrices.

$$Q_1, Q_2 = W_1^Q F_1, W_2^Q F_2$$

$$K_1, K_2 = W_1^K F_1, W_2^K F_2$$

$$V_1, V_2 = W_1^V F_1, W_2^V F_2$$

Cross-attention between them was computed using the normalized dot product to obtain interaction information.

$$F_{1\to 2} = \text{softmax}(Q_2 K_1^T / \sqrt{d}) V_1$$

$$F_{2\to1} = \text{softmax}(Q_1 K_2^T / \sqrt{d}) V_2$$

After this step, and similar to normal transformers, the concatenation, layer normalization, and feed-forward network operations proceed as expected except they are applied to both feature sets. In the final step, the two cross attended features are concatenated together into one feature set [28]. They found that this type of fusion allowed the final feature set to significantly outperform each individual feature set. Many examples of different variations of cross feature fusion utilizing different input types appear in many state-of-the-art SER tasks including hierarchical cooperative attention [21]; HuBERT, MFCC, and prosodic features for multi-lingual SER [58]; Wav2Vec 2.0, HuBERT, and spectrogram features [55]; and audio and text features [27]. In these examples, cross attention feature fusion captures the interaction between the sequential features to obtain complementary learned properties in the final feature set thereby improving performance compared to the individual features themselves.

In MER, cross attention fusion has not been as widely used, however, in cases where it has been applied, it has also been shown to improve performance. Huang et al. applied this approach to fusing symbolic MIDI data, Mel Spectrograms, and high-level features. They used both the DEAM and PMEmo datasets for regression evaluation, resulting in RMSE scores of 0.102 and 0.097 for valence and arousal respectively. For classification, a Macro-F_1 score of 66.2% was obtained on the EMOPIA dataset [19]. Finally, in a separate study, Shi et al. reported a 3.55% improvement in F_1 when incorporating handcrafted music theory representations

with audio and lyric modalities [48]. Beyond these initial attempts, more studies need to be conducted to explore how cross-feature fusion can integrate different modality representations.

Chapter 4: Dataset

In our work, we utilize the Music Emotion Recognition next GEneration, or MERGE, dataset created by Louro et al [34]. Created in 2021, this dataset was designed to be one of the largest, publicly available music emotion recognition datasets with quality controlled labels. The dataset contains 3,554 distinct audio samples, each thirty seconds in length, and each song is discretely labeled with one quadrant of Russell's Circumplex Model of Emotion [45].

After an extensive review, Louro et al. determined that there were a number of issues with many of the other MER datasets. Specifically, with the larger datasets they found significant diversity in the emotional taxonomies used to label the datasets, most of the larger-scale datasets are annotated through social tagging or automatic methods without human validation, there is little variation in the types of music represented, only song IDs with no audio were provided, and there were no collective baselines. MERGE was developed to address these issues while maintaining a relatively large size and meeting the following requirements: a valid taxonomy; variety and balance across musical genres, styles and eras; careful annotation; reduced ambiguity in annotator agreement; publicly available; inclusion of baseline scores; and accompanied by a rich set of metadata.

To create this dataset, audio clips were obtained from the AllMusic API. In addition to the audio, the API provides 289 distinct emotion tags. Using Warriner's

Dataset	Q1	Q2	Q3	Q4	Total
MERGE Audio Complete	875	915	808	956	3554
MERGE Audio Balanced	808	808	808	808	3232
MERGE Bimodal Complete	525	673	500	518	2216
MERGE Bimodal Balanced	500	500	500	500	2000

Table 4.1: The distribution of audio samples between the different versions of the MERGE dataset. Recreated from [34].

adjective list [50], which provides a mapping from 13,915 adjectives to the valence and arousal space, they were able to map each song's emotional tags to a single point in Russell's model. To reduce ambiguity, songs whose labels were close to the origin were removed. Next, the authors conducted a blind inspection of the audio to ensure the quality of the annotations. Participants in this inspection were randomly assigned songs and asked to annotate them in the Valence-Arousal plane. Of the eight possible human annotations, a song was removed if none matched the automatic mapping. Additionally, noisy, silent, or otherwise poor quality songs were removed [34]. The 3,554 total remaining songs represent the largest human-validated publicly available MER dataset.

The 3,554 remaining songs comprise the MERGE Audio Complete version of the dataset, and this is the dataset we use in this study. The distribution of labels can be seen in Table 4.1. In addition, other versions were also created, such as the Audio Balanced version which contains 808 songs per quadrant. Finally, the Bimodal Complete and Bimodal Balanced versions contain not only raw audio but also song lyrics. We choose not to use these as they contain fewer total samples and our experiments focus solely on audio.

Input Type	**Modality**	**Macro F_1**	**Weighted F_1**
Handcrafted features	Audio	70.12	–
Handcrafted + Spectrogram	Audio	68.84	–
MERT embeddings	Audio	–	72.37
Lyrics + Spectrogram	Audio + Lyrics	–	79.21

Table 4.2: Baseline performance metrics for the MERGE dataset provided by [34] [33] [35].

The baselines for the MERGE dataset that are relevant for this study can be seen in Table 4.2. The initial baseline, 70.12% F_1-score, was reported in [34] and was obtained using handcrafted musical features evaluated with an SVM with an RBF kernel function. These handcrafted features are the same static 1,704 features used in their previous work [41] with the smaller 4QAED dataset, on which they achieved comparable results. In two separate publications, Louro et al. further increased their performance on this dataset in [33] and [35]. In their first paper, they utilized the same 1704 handcrafted feature set in addition to learned features extracted from mel spectrograms with a CNN network. Late-stage fusion was used to concatenate these features together, and then two dense classifier heads were used to produce their final output. To pre-train the CNN, synthetic audio samples were first used before freezing the model's weights. With this architecture, they were able to achieve 77.62% F_1 on the balanced version of the dataset, but only 68.84% F_1 on the complete dataset. This was due to poor discrimination between the Q3 and Q4 classes (songs with low arousal) which is a common issue reported by many in MER studies [33].

In their second study [35], they did not use handcrafted features, and instead

explored the bimodal aspect of the dataset, utilizing both audio and lyric embeddings obtained from pre-trained models. For the audio modality, they chose to use Wav2Vec 2.0, MERT, and the also the same Mel Spectrogram CNN feature extractor from their previous work [33]. To obtain static embeddings from the pre-trained models, they utilized a one-dimensional convolutional layer to reduce the hidden-layer dimension and then average the time steps to obtain a single 1×1024 feature vector for each song. For their lyric embeddings, they chose RoBERTa, ModernBERT, and their own CNN network. When these two modalities were utilized together, the best result, 79.21% F_1, was obtained using the audio spectrogram model and their lyric CNN network. The best result using the pre-trained models was 77.93% F_1, obtained through majority voting across all models to obtain a prediction [35]. For our purposes, since we are not examining lyrics, their audio-only results are particularly relevant. Interestingly, the audio CNN feature extractor achieved only a 62.10% F_1. Their best audio-only result, 72.37% F_1, was achieved on the complete dataset using the MERT model [35]. Finally, for all of these experiments, Bayesian optimization with early stopping was utilized to find hyperparameters, such as batch size, optimizer type, learning rate, and number of units.

Chapter 5: Methods

In this chapter, we describe our process for extracting the features and embeddings from raw audio. These are later used in our experiments to classify emotion in the MERGE dataset.

5.1 Handcrafted Features

We extract a total of 710 handcrafted acoustic features from five separate audio toolkits: Essentia [6], MIR TB [25], PsySound3 [8], Yaafe [39], and Marsyas [49]. These handcrafted features are mostly low and mid-level features with a few high level features like danceability, and are the same ones used in [29]. The same method for extraction is employed here. Before extraction, we resample our audio to a sample rate of 16 kHz, and pad or clip all songs to exactly 30.0 seconds. This results in 480,000 total samples for each song. Similar to our previous work [29], we utilize a frame window of 2048 samples and a hop size of 1024 samples. These are default values for a majority of the extractors with the exception of PsySound3 which does not split its audio by default. To combat this, we manually split our audio into the required frames before they are passed to PsySound3 and manually combine their values in post-processing. With these parameters, this results in a total of 467 time frames for the sequential features of each song. Essentia and

Musical Dimension	Feature Count
Melody	14
Harmony	99
Rhythm	46
Dynamics	63
Tone Color	449
Expressive	6
Form	2
High-Level	1

Table 5.1: The distribution of our handcrafted features by musical dimension.

Yaafe are both Python3 based, and we utilize them to extract 220 and 95 features respectively. MIR TB and PsySound3 are MATLAB based, and are used to extract 100 and 77 features. Finally, Marsyas is a command-line tool which was used to extract a total of 218 features. The exact features are described in more detail in [29].

The features we extract can be categorized into eight different musical dimensions: Melody, Harmony, Rhythm, Dynamics, Tone Color, Expressivity, Form, and High-Level. This organization is common in music processing and can be seen in [42] and [41]. All eight of these dimensions have been shown to affect emotion in music in some manner.

Melody is the horizontal sequence of pitches perceived in a song that forms the main motif of the song. It is often regarded as one of the most memorable and recognizable elements in a song. Research has shown the pitch, as well as the variation in the pitch of a song that defines the melody, has shown to have a strong effect on the emotion. For example, Gabrielsson and Lindström found that songs

with higher frequencies and greater pitch variation lead to higher arousal values and emotions like joy and anger [17]. Additionally, Balkwill and Thompson found that songs with complex melodies are more often associated with negative valence [5].

In contrast, harmony describes the vertical aspect of a song, or the combination of different notes to form chords. Similar to melody, chords that contain higher frequencies are often linked to positive emotions such as joy and euphoria. However, Gabrielsson and Lindström also found that songs with unchanging harmonies evoke stable feelings like calm and relaxing emotional states which correspond to low arousal values. Alternatively, songs with complex and frequently changing harmonies are more associated with anxiousness and negative valence values. Finally, musical scale significantly influences the emotion where songs written in a major scale correspond to happy and bright emotions, whereas minor scale songs sound darker and have a sadder or more serious tone [17].

In addition to melody and harmony, rhythm is considered another central contributor to the emotion of a song, and some researchers consider it the most important aspect [22]. Most songs are composed of repetitive musical notes where these notes last a certain amount of time and occur at a common rate. Rhythm describes these repetitions with features like tempo which describes how often these repetitions occur, and other features denote how they can change over time. When examining how rhythm influences emotion, Fernándex-Soto et al. found that faster tempos can be linked to higher arousal emotions like happiness and surprise. Additionally, they found that songs with varying tempos can elicit specific emotions

[16]. These results agree with Gabrielsson and Lindström's findings that songs with regular rhythm patterns were associated with positive-valence emotions such as happiness and peacefulness. Complex, changing, and irregular patterns had the opposite effect, leading to emotions such as anger and unease [17].

The dynamics dimension in music describes the energy level and loudness of the notes that are played, and how these values change over the course of the song. As one may expect, music that contains loud instruments leads to higher arousal values with intense emotions like joy and anger. On the contrary, softer notes and songs are related to emotions like calm, serene, and sad [22]. When studying how dynamics change over the course of a song, Gabrielsson and Lindström found that an increase in the variation of loudness can have a negative effect on the valence dimension [17].

Tone color, or timbre, is the musical dimension which describes the sounds which make individual instruments unique. As such, this particular dimension can only be quantified with lower-level descriptors which represent characteristics of the instrument's sound waves and spectral composition. These low level features have frequently been used to differentiate among different instrument sources such as separating vocals from background instruments, or distinguishing between string and wind instruments. In an investigation into whether emotional associations stem from the inherent sound of an instrument or how it is used in the song itself, Wu et al. found that many instruments are naturally predisposed to evoke certain emotions. For example, they reported that the horn and flute instruments most effectively conveyed the emotions of sadness and depression [51]. They further

demonstrated that spectral features such as spectral centroid and odd to even harmonic ratio are closely linked to these emotional associations.

These findings supported earlier claims from Gabrielsson and Lindström who found that instruments with sharper amplitude envelopes are associated with more positive valence emotions, whereas those with rounder envelopes are linked to negative emotions such as disgust and sadness. Finally, they found a relationship between the number of harmonics and emotion, in which fewer harmonics can lead to emotions like boredom, happiness, or sadness [17].

The expressivity and form dimensions represent more complex descriptions of music, and contain fewer features. Instead, many of them require the MIDI representation of a song, so we are limited to only those which can be extracted from the raw audio. Additionally, compared to the other dimensions, fewer studies have investigated the relationship between them and emotion.

Expressivity describes how rhythm, dynamics, and articulation interact within a song [41]. Rather than focusing on what notes are played and when, it reflects how the musical elements are executed. For example, staccato articulation, characterized by short and detached notes, has been associated with higher arousal values [17].

The form dimension refers to the overall structure and organization of a song. It describes how a song is laid out and divided into sections, or frames, and how these sections influence one another. Relatively few studies have examined the impact of form on emotion, however, Balkwill and Thompson found that songs with simpler structures tended to have more positive emotions, and more complex

structures with negative ones. However, no specific relation with the arousal level was found [5].

Finally, high level features are those that cannot be included in any of the other dimensions and include features like danceability from Essentia.

After extracting all 710 features, it is necessary to post-process some of these features for a number of reasons. First, many of the features (e.g., MFCCs) are output as multidimensional matrices of shape (n, 467) to represent bins of frequencies. Instead of treating these as single features, they are split into n separate features. For features that are inherently static, or global, only a single value is output. For example, a song has one beats-per-minute (bpm) feature. These single values are repeated across the 467 time steps. While this means that these features are not dynamic, there are very few of them, and from [29], features like bpm were shown to be extremely important. For features such as key and chord progression, which are output as strings, we utilize the Circle of Fifths to convert them to numerical values. We represent A minor through D minor as the values 0-11, and C through F as values 12-23. Finally, certain features like onsets and scale are represented as binary vectors representing the presence of an onset, and minor or major mode.

As a final post-processing step, we remove all features that remain constant over all time steps and over all songs. For features whose absolute correlation exceeds 0.95, we remove one feature and retain the other. This leaves us with a final feature set of 608 features $\times$ 467 time steps. For our static handcrafted feature set, seven statistical aggregates (mean, median, min, max, std, skew, kurtosis) are

calculated across the time dimension, and we utilize the same top 300 features from the results of our feature selection in [29].

5.2 Pre-Trained Models

In our experiments, we utilize three different pre-trained models to extract embeddings from our audio: specifically, the Wav2Vec 2.0, MERT, and MAEST models. These models differ in their architectures, training objectives, and inputs.

5.2.1 Wav2Vec 2.0

Developed by Meta, the Wav2Vec 2.0 model was originally created as a successor to their first model for speech recognition, and is one of the most widely used audio models. At a high level, it utilizes self-supervised contrastive learning to create a meaningful embedding space which is then used for speech classification. Its pre-training pipeline is as follows. The raw audio is used as input for 7 convolutional feature encoders to obtain latent speech representations. These feature encoders perform convolutions along the temporal dimension of the audio followed by layer normalization and the GELU activation function. These latent feature representations are then used in two ways. Firstly, they discretize them into a finite set of speech representations using a product quantization module which was developed in prior work. This new set of speech representations is then used as the targets for the self-supervised objective. The second way they use the latent

feature encodings is as input into a traditional transformer architecture to obtain contextual representations, and in order to obtain positional embeddings, they use a convolutional layer instead of fixed values. This process can be seen in Figure 5.1. During pre-training, a proportion of time steps is masked in the feature encoding, and the training objective is to correctly identify the quantized representation for each time step using the contextual representations. In order to solve this, they utilize a contrastive loss function which measures the cosine similarity between the contextual representations and the quantized representations. Finally, after pre-training, they fine-tune the model on labeled Librispeech data by adding a linear projection to the contextual representations. They found that their model outperformed many other semi-supervised methods while being much simpler [3]. While this is the most used audio model, it is important to consider that their speech recognition task differs from music-related task, and the model was not trained on any music.

For our experiments, we use the Wav2Vec 2.0 model from HuggingFace. Specifically, we use the large version of the model which was trained on 960 hours of 16kHz audio [3]. The model outputs 1,024 contextual features across 2249 time steps and 25 hidden layers. We follow a similar process as [35] in which we utilize a one-dimensional CNN layer to reduce the 25 hidden layers, and we utilize one dimensional average pooling and linear interpolation in order to reduce the time steps to the same number as our handcrafted features yielding a dynamic input of shape 467×1024 for a single song. In order to get static features, we completely reduce the time steps by averaging them together into a single value leaving us

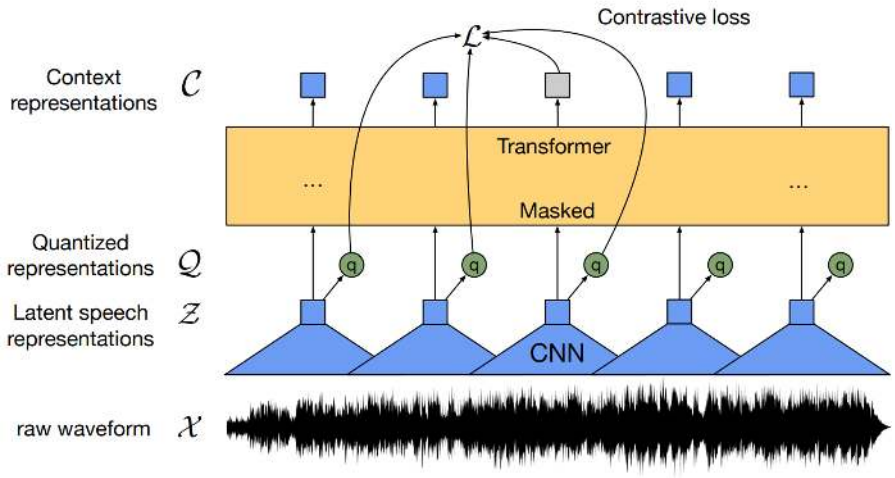


Figure 5.1: A visualization of the Wav2Vec 2.0 framework which learns both contextualized speech representations and quantized speech units. Figure from [3].

with a single 1,024-dimensional vector per song.

5.2.2 MERT

The second pre-trained model we test is MERT, or the acoustic Music undERstanding Model with large-scale self-supervised Training. As the name suggests, this model is trained specifically for music tasks instead of general audio or speech. This is in an attempt to model musical characteristics such as tonality and pitch. Like Wav2Vec 2.0, it is also a self-supervised model which generates pseudo targets for sequential audio clips. To generate these pseudo targets, they tested two acoustic teachers and one musical teacher. The two acoustic teachers used k-means on the log-Mel spectrum and Chroma features to encode timbre and harmonic features, and a Residual Vector Quantization-Variational AutoEncoder (RVQ-VAE)

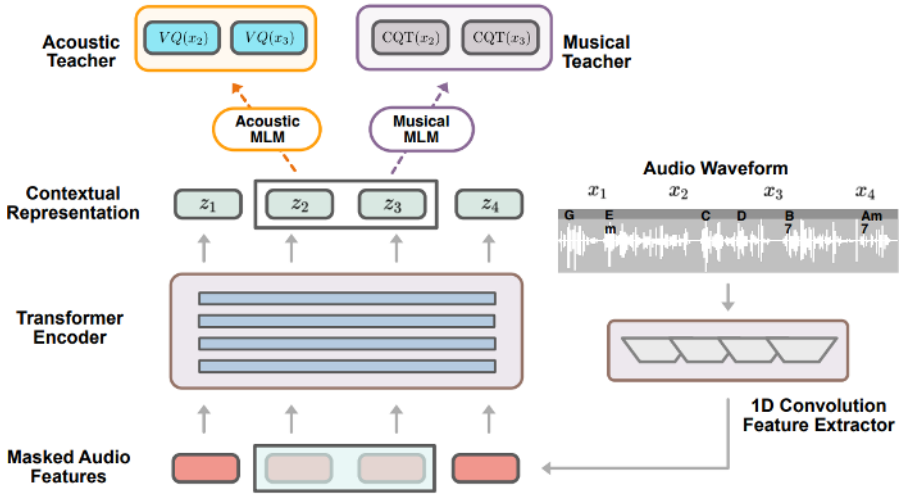


Figure 5.2: A visualization of the MERT pre-training framework which utilizes a student teacher framework with acoustic and musical teachers. During pre-training, the model attempts to reconstruct masked time steps to maximize teacher clustering and MSE. Figure from [32].

to model timbre. They used the Constant Q Transform (CQT) to model music information, which is a commonly used frequency transform in MIR, and emphasizes pitch and harmonic features. In their experiments, they found a linear combination of RVQ-VAE and CQT to perform the best. Their model is as follows. Raw audio is used as input to a one dimensional CNN to extract features. A proportion of these features are then masked across the temporal dimension and fed into a transformer architecture to obtain contextual representations. These contextual representations are used to generate pseudo-labels using the two teacher models. Their pre-training objective is to reconstruct the masked time steps in a way that maximizes the log-likelihood of belonging to a similar cluster for the acoustic features, while simultaneously minimizing the mean squared error of the CQT recon-

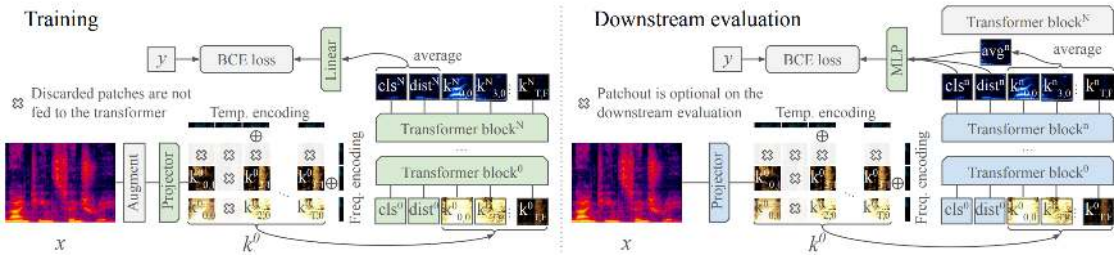


Figure 5.3: A visualization of the supervised training and downstream evaluation procedure of MAEST. Trainable blocks are shown in green and frozen blocks in blue. Figure from [2].

struction. This process can be seen in Figure 5.2. On a wide range of downstream MIR tasks such as instrument tagging, genre, general tagging, rhythm, emotion recognition, key detection, and pitch tasks, they achieved SOTA or comparable results even when training on less data [32].

Like Wav2Vec 2.0, this model is available through HuggingFace, and also outputs 1024 contextual features across 2249 time steps and 25 separate layers. Additionally, we use the largest v1-330M version which is pre-trained on 160 thousand hours of unlabeled music. In the same manner as the Wav2Vec 2.0 features and [35], we utilize a one dimensional CNN layer to reduce the hidden layers, and average pooling and linear interpolation to obtain dynamic features of shape 467×1024 for each song. The time dimension is similarly averaged into a single value for the static version of the features as well.

5.2.3 MAEST

Finally, the last pre-trained model we use is Essentia's MAEST, or the Music Audio Efficient Spectrogram Transformer. This model differs from both Wav2Vec 2.0 and MERT in a number of ways. Firstly, it is a supervised model instead of using self-supervision for pre-training. Additionally, instead of using raw audio as input, it uses a Mel Spectrogram, a visual representation of the audio. Finally, it does not use any convolutions in its model and instead uses 12 transformer blocks. Their Mel-Spectrograms are first linearly projected and split into 16 by 16 patches with 10 by 10 strides which represent the different frequencies of the audio at different time steps. These patches are then combined with their respective frequency and temporal encodings. Patchout is used similarly to masking in a way which proportionally removes entire rows (frequencies) or columns (time steps) during training. The remaining patches are prepended with classification and distillation trainable tokens, and are input into the 12 transformers. They used an in-house dataset comprised of four million songs whose labels were obtained by mapping to the Discogs' top 519 music-style multilabel taxonomy. Since this is a supervised model, the transformer tokens are used to directly classify these tags during training. However, during training only the average of the classification and distillation tokens are used for prediction and to calculate their loss. The average of the learnable sequential tokens is used only during downstream evaluation, in conjunction with the other two tokens. When testing on other downstream tasks, the transformers' weights were frozen and an MLP was used for other tagging tasks such as

genre, instrument, mood, and general tagging. Through their experiments, they found the seventh hidden layer to provide the most informative representations for these downstream tasks, and reported several SOTA scores on many tasks [2].

For our purposes, we use all 768 learnable sequential tokens in addition to the learnable classification and distillation tokens (3×768) from the seventh layer for our experiments. We use the 30s-pw-519l version which they found to be the best performing version in most downstream tasks. Since we already know the best-performing layer, do not use a one-dimensional CNN layer, and only pool the time steps. For our dynamic set, we obtain features of shape 467×768 in which the classification and distillation tokens are prepended to the sequential tokens. For the static set, all tokens are averaged into a single 768-dimensional vector.

Chapter 6: Experiments

In this section, we describe the experiments conducted using the features extracted from the MERGE audio dataset. We compare the performance metrics of these experiments against the baseline metrics for the dataset [34] as well as the subsequent papers [33] [35]. The metrics we use include macro and weighted F_1-scores for classification tasks and Root Mean Squared Error and Pearson's Correlation Coefficient (r) for regression tasks. Macro F_1 is defined as:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

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

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

Whereas the macro F_1-score weights all classes equally, since we use the Complete MERGE dataset, the class distribution is imbalanced. In order to proportionally weight each class, the weighted F_1-score is calculated as a weighted sum of the per-class F_1-scores, where the weights are determined by the proportion of the class distributions. For regression tasks, RMSE measures the prediction accuracy and fit of the predicted values versus the true values. It is defined as $\sqrt{\frac{1}{n}\sum^{n}(y_i - \hat{y_i})^2}$ and lower values indicate better predictions. Pearson's correlation coefficient measures

the strength and direction of a linear relationship between two continuous variables, and is defined as $r = \frac{Cov(X,Y)}{\sigma_x \sigma_y}$ For all of our experiments, the dataset is split into ten folds, and our performance metrics are obtained from 20 independent seeds using ten-fold cross-validation. For this cross validation, eight folds are used for training, one for validation, and one held-out test fold. Additionally, we employ Bayesian optimization on 50 trials to select the optimal hyperparameters for batch size, learning rate, and other model specific parameters. In order to reduce computation time, five folds are used instead of ten for hyperparameter search, with four folds used for training and one for validation. The validation loss for all five folds is averaged to obtain a single validation loss score for that set of hyperparameters. We also utilize early stopping with a patience value of 15 to select the model parameters which minimize our validation loss for each task. If the validation loss continuously increases for 15 epochs, then training is stopped before the maximum number of epochs is reached, and the model weights are restored to their values at the lowest validation loss.

6.1 Handcrafted Feature Optimization

In our previous work [29] on static features obtained through statistical aggregation, we found that feature selection was a pivotal step in classification. With over 3000 statistical features, reducing this amount to the most emotionally informative features greatly helped increase our performance. The method we found most effective was a combination of L1 regularization to reduce to 800 features,

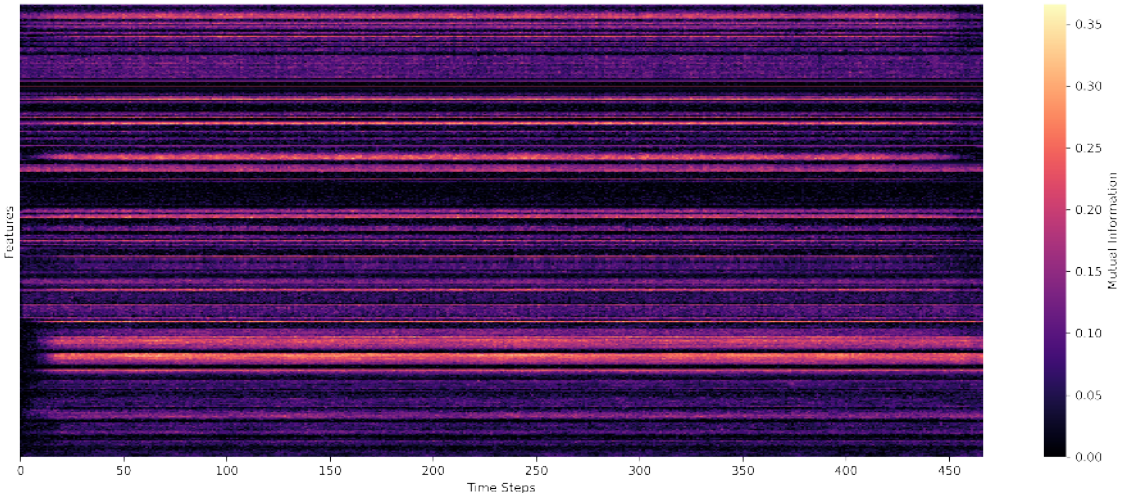


Figure 6.1: A visualization of the mutual information for each feature over time.

and then recursive feature elimination to reduce to the final 300. However, with the increased dimensionality of our sequential features, L1 regularization is not possible and recursive feature elimination is not computationally feasible within a reasonable amount of time.

Our solution is to utilize mutual information over time; specifically, we compute the mutual information of all features at each timestep. A visual representation of this can be seen in Figure 6.1. With these values, we calculate the mean mutual information over time for each feature, and remove those with average values below 0.05 and 0.1, resulting in 430 and 225 total features, respectively.

In a separate test, we used a learnable L1 regularization layer to mask feature inputs. This mask layer consists of learnable weights w which are randomly initialized with a uniform distribution between 0.01 and 0.05. This layer is used as the initial layer in the model in order to gate (i.e. $w = 0$) features. During the

Feature Amount (d)	**Macro F_1**
All (608)	**67.97**
MI (430)	66.70
MI (225)	65.60
Feature Mask	66.93
PCA (200)	66.82
PCA (400)	67.19

Table 6.1: Results using different feature selection and dimensionality reduction techniques for our handcrafted set of features.

forward pass, each input feature is scaled by the nonnegative scalar m_i, $y_i = x_i m_i$ for $i = 1, ..., n$, where n is the number of features. With L1 regularization and cross entropy loss, our loss function becomes:

$$L(y, \hat{y}) = -\sum^{N} y_i log(\hat{y_i}) + \lambda \sum^{k} |w_j|.$$

With small initial weight values, the model learns which gates to open, or increase, encouraging sparsity in m_i with regularization.

Finally, Principal Component Analysis (PCA) was also tested as a dimensionality reduction method. The chosen dimensionalities were 200 and 400.

At the static level, feature selection provided a boost in performance; however, that is not the case here as seen in Table 6.1. The best results were obtained using all 608 features, however, the different feature selection methods only decreased performance marginally. Because of this, we do not remove any features from our handcrafted feature set in the later experiments in this paper.

6.2 Dynamic vs. Static

As described in Section 5.2, we test our input features from three different pre-trained Large Audio-Language Models (Wav2Vec 2.0, MERT, MAEST) and our own handcrafted feature set. To obtain static global features from the Large Audio-Language Models, we first extract 4D tensor features from raw audio, or in the case of MAEST, Mel-Spectrograms. The dimensions of these 4D tensors correspond to $n \times l \times t \times f$ where n is the amount of audio samples, l is the amount of hidden layers, t is the number of time steps, and f is the feature count, and our goal for static feature input is $n \times f$. In order to obtain this, we follow operations similar to those performed by Louro et al. in [35]. We first reduce the hidden layers by employing a 1D CNN that learns which layers contain the most relevant information. Then we average the time steps together into a single feature value. This process works for both Wav2Vec 2.0 and MERT, however, the process for the MAEST model is different. Instead of reducing the hidden layers with a 1D CNN, we obtain only the features from the seventh hidden layer. In their experiments, Alonso-Jimenez et al. found this layer to perform the best for downstream tasks [2]. It is also important to note that this model is trained with classification and distillation tokens prepended to the learnable sequence tokens. For our dynamic experiments, we only use the sequential tokens ($N \times 465 \times 768$) as input into our recurrent model, and later concatenate the other two tokens for input into our classification head. For our static experiments, we test both averaging all tokens together and a separate experiment with only the classification tokens. For static features we use

2 fully connected layers with dropout in between and a classifier layer. Similar to our approach with dynamic features, we use Bayesian optimization to determine the best hyperparameters like batch size, learning rate, and number of dense units.

6.2.1 Dynamic Results

For our dynamic features, we employ a model that builds upon architectures that have historically performed well from [14] and [9]. Our model is comprised of two 1D CNN layers followed by two BLSTM layers. For our CNN layers, we utilize ReLU activation functions, and for the BLSTM layers, we use tanh. After this, we utilize attention pooling to determine the most important time steps. This pooling removes the time dimension, allowing a fully connected classifier head with the softmax activation function to determine the most probable class. In addition to following previous models that worked well, the relatively small size (3,554 songs) of our dataset limited how strong our model could be, and our embeddings are extracted from models which themselves employ transformers. In order to show that our model is best able to extract meaningful representations from these features, we also compare our results with models trained with one transformer block, two transformer blocks, and a time distributed fully connected model.

For our exact hyperparameters, we use kernel sizes of 3 and 5 for the CNN layers in order to gradually increase their receptive field, and a stride of 2 for the first CNN layer to reduce the number of time steps by half. Finally, through initial testing, we found the optimal number of units for the BLSTM layers to be 128.

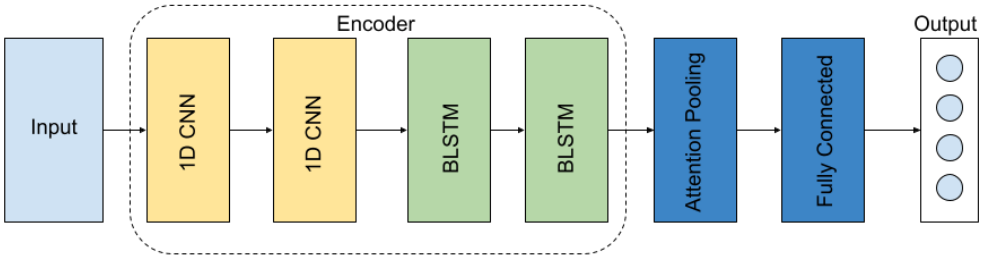


Figure 6.2: Our standard dynamic model. The Encoder layers inside the denoted dotted line are reused in further figures.

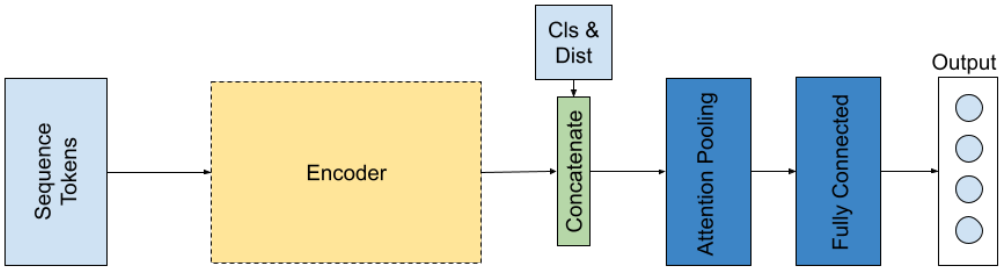


Figure 6.3: The model utilized for the MAEST embeddings. Only the sequence tokens are input into the encoder block and concatenated with the classification and distillation tokens before pooling.

Feature Type	**Macro F_1**	**Weighted F_1**
Handcrafted	67.97	68.29
Wav2Vec 2.0	66.78	67.10
MERT	73.48	73.70
MAEST	**76.52**	**76.72**
MAEST sequence tokens	76.37	76.57

Table 6.2: Results of our experiments using dynamic features extracted from handcrafted methods and three pre-trained models.

From the results of this experiment shown in Table 6.2, the best-performing features were those extracted from the MAEST model followed by MERT, our handcrafted features, and finally Wav2Vec 2.0 close behind. It is also important to note that the MAEST model, which utilized the classification and distillation tokens in addition to the sequence tokens, slightly outperformed the sequence tokens alone. However, since it only slightly outperformed, this leads us to believe that a majority of the information was derived from the sequence tokens with a slight increase from the classification and distillation tokens.

Similar to [37], it is unclear what the optimal frame size should be. In order to test this, we downsample the MAEST features from the original 467 frames to both 60 and 30 total frames. With our 30 second audio clips, this yields 0.5 and 1 second per frame, respectively. These downsampled embeddings are obtained utilizing `scipy`'s resample function. Importantly, since MAEST has classification and distillation tokens, only the sequence tokens are downsampled to the target length, and then the classification and distillation tokens are prepended as extra time steps. Similar to our original model, only the down-sampled sequence tokens

Number of Frames	Seconds Per Frame	F_1
467	0.064	76.72
60	0.5	76.53
30	1	76.63

Table 6.3: Results from downsampling our MAEST embeddings to 60 and 30 time steps.

Model	Macro F_1	Weighted F_1
One Transformer Block	76.53	76.75
Two Transformer Blocks	76.40	76.62
Time-Distributed FC	75.52	75.73

Table 6.4: Results from employing other dynamic models with MAEST embeddings.

are used in the CNN-BLSTM layers, and the other tokens are concatenated before attention pooling. For this experiment, we utilize the same model for all three conditions and only change the amount of time steps. As you can see in Table 6.3, the 467 time steps produce the best F_1, however marginally. Based on these results, we do not apply further downsampling in subsequent experiments.

To show that our model is creating emotionally relevant representations, we compare it with transformer models with one and two blocks, and a time distributed fully connected model. Similar to the CNN-BLSTM model, all of these models utilize attention pooling before the classifier head. Our CNN-to-BLSTM outperforms all three, as seen in Table 6.4.

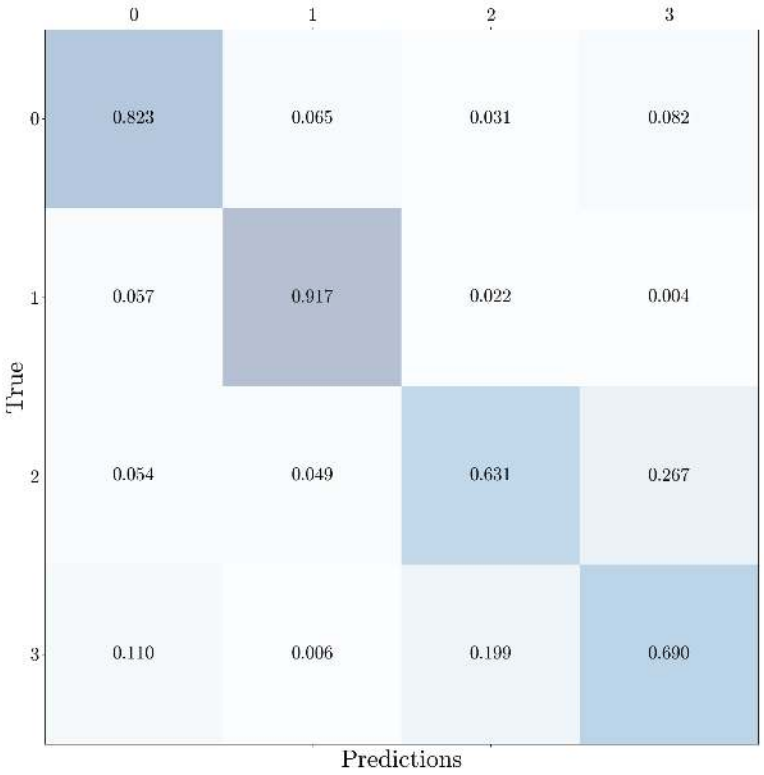


Figure 6.4: This confusion matrix shows the distribution of our classification results when utilizing the dynamic MAEST embeddings.

6.2.2 Static Features

For our static experiments, we utilize the same three pre-trained models and our handcrafted feature set. In order to obtain static features, we simply average over the time domain, as was done in [35]. This includes the classification and distillation tokens for MAEST, however, we also test just the classification tokens. For our static handcrafted feature set, we utilize the same 300 statistical aggregate set as in [29]. We utilize a fully connected MLP model with two hidden layers and

Feature Type	**Macro F_1**	**Weighted F_1**
Handcrafted	68.39	68.75
Wav2Vec 2.0	68.27	68.55
MERT	74.51	74.71
MAEST avg	**77.57**	**77.78**
MAEST cls	75.98	76.20
Handcrafted [34]	70.12	–
Handcrafted + Spectrogram [33]	68.84	–
MERT [35]	–	72.37
Lyrics + Spectrogram [35]	–	**79.21**

Table 6.5: Results from our static features can be seen in the top half of the table. The handcrafted features are the same set from [29], and the features from pre-trained models are obtained by averaging their temporal dimension. The scores on the bottom half reflect the baseline metrics for the MERGE dataset, also obtained with static representations.

an output classifier.

As seen from the results of our static experiments in Table 6.5, we see a similar pattern to the dynamic results. The MAEST embeddings, both averaged and classification, produce the highest scores, followed by MERT, handcrafted features, and finally Wav2Vec 2.0. However, the static experiments produce slightly higher F_1-scores across all feature types with the average MAEST embeddings producing the highest at 77.57 macro F_1 and 77.78 weighted F_1.

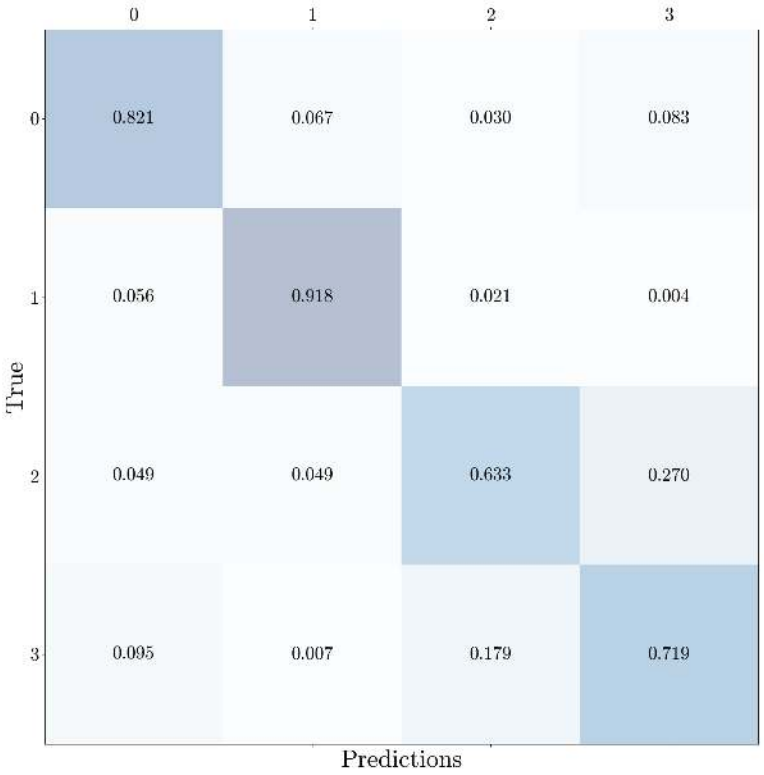


Figure 6.5: This confusion matrix shows the distribution of results from our static MAEST embeddings, or our best results.

6.3 Fusion

As was seen in [33], [35], [27], [19], and [48], feature fusion can improve the quality and variety of inputs, resulting in better representations and performance. For our dynamic experiments, we test three different fusion strategies: early, late, and mid-level.

We utilize the same CNN-BLSTM model as in the previous experiments, and study the performance resulting from combining all four feature types. For early

fusion, the inputs are concatenated along the feature dimension by aligning the time steps across feature types. With our four feature types (Handcrafted (608), Wav2Vec 2.0 (1024), MERT (1024), and MAEST (768)), this would result in a total input matrix of (467, 3424). In order to reduce this amount, each feature type is first projected to 448 dimensions using a fully connected layer, resulting in a final input matrix of (467, 1792). After this concatenation, the entire feature space is used as input to the CNN-BLSTM + attention network.

As opposed to early fusion, in late fusion each separate feature type is input into separate CNN-BLSTM + attention encoders, and they are concatenated only after attention pooling. This allows each encoder block and attention pooling to learn representations independently for each feature type. Then a fully connected classifier is applied to the combined features for final classification.

Finally, we use cross attention for mid-level fusion. Similarly to late level fusion, each input utilizes its own encoder block, but there is no attention pooling. This results in sequential representations from each encoder which we then utilize for our cross attention fusion. Our cross attention layer is similar to the one used in Section 3.2 in which the softmax attention values are obtained utilizing the Key and Value matrices from the same feature vector and the Query matrix is obtained from the other. We try two different variations of this feature fusion. One in which the features from the handcrafted set are cross-attended to the other three feature embeddings, and one in which all features are cross-attended with all other features. We make this distinction since the handcrafted set is not obtained from a pre-trained model, and it contains more low-level features which may not be

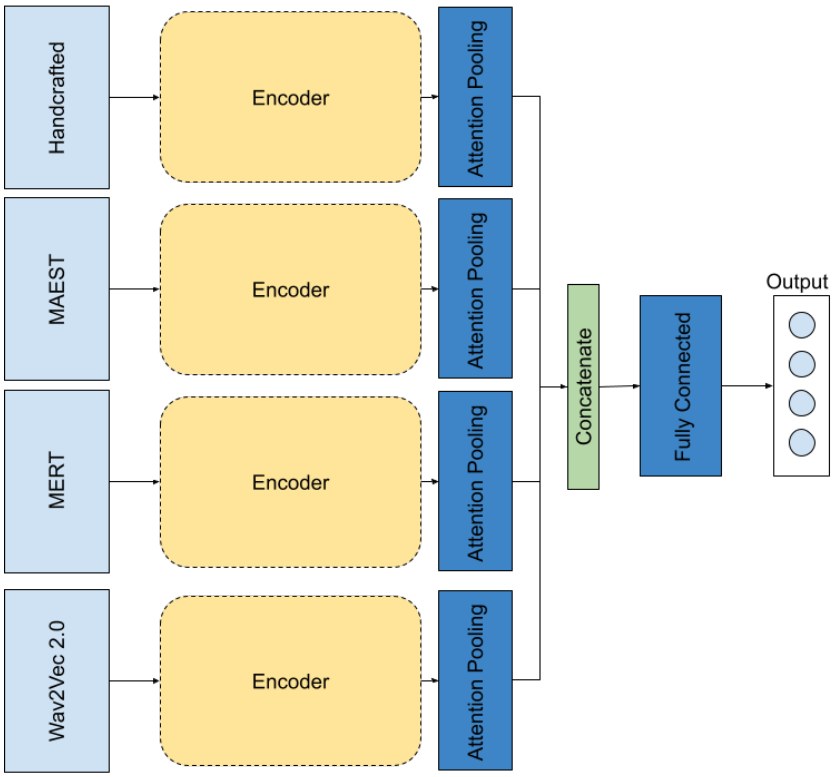


Figure 6.6: Our late fusion architecture which combines the four separate inputs.

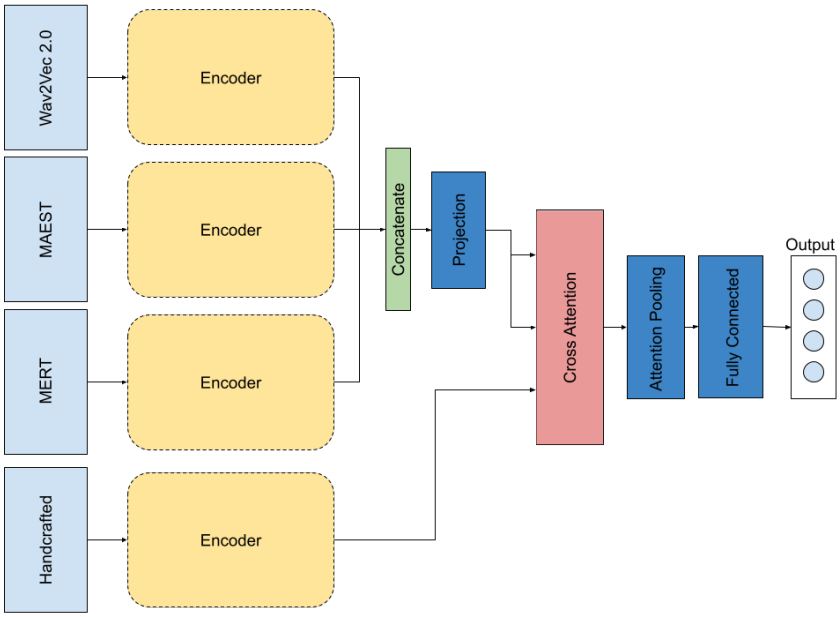


Figure 6.7: Our mid-level fusion model which utilizes cross attention to fuse inputs.

present in the embeddings. The results from our dynamic feature fusion methods can be seen in Table 6.6.

As you can see, late and early fusion both slightly outperformed the best-performing dynamic MAEST embeddings (the top performing dynamic input type), but the cross attention methods did not. However, none of the dynamic fusion

Fusion Type	**Macro F_1**	**Weighted F_1**
Late	76.95	77.14
Early	**76.72**	**76.91**
Cross (Hand - embeddings)	74.45	74.66
Cross (all - all)	74.55	74.75
Dynamic MAEST	76.52	76.72

Table 6.6: Results from different fusion strategies on our dynamic features.

Fusion Type	**Macro F_1**	**Weighted F_1**
Early	76.72	76.92
Late	77.16	77.34
Static MAEST	**77.57**	**77.78**

Table 6.7: Results of different fusion strategies on our static features.

methods were able to outperform the static MAEST embeddings.

We perform comparable fusion experiments with our static feature set as well, however, since there is no temporal dimension, cross-attention fusion is not possible. Nonetheless, early fusion works much the same for the static features. All feature types are concatenated together into a single 3,100-dimensional vector and input into a MLP with two hidden layers.

Late fusion works similarly, but instead of CNN-BLSTM encoders, each feature type is separately input into two fully connected layers. Afterward, their representations are concatenated and input again into two fully connected layers with the final layer serving as the output layer. We utilize the same layer widths for the separate hidden layers as their respective individual networks. The results from our static fusion experiments can be seen in Table 6.7.

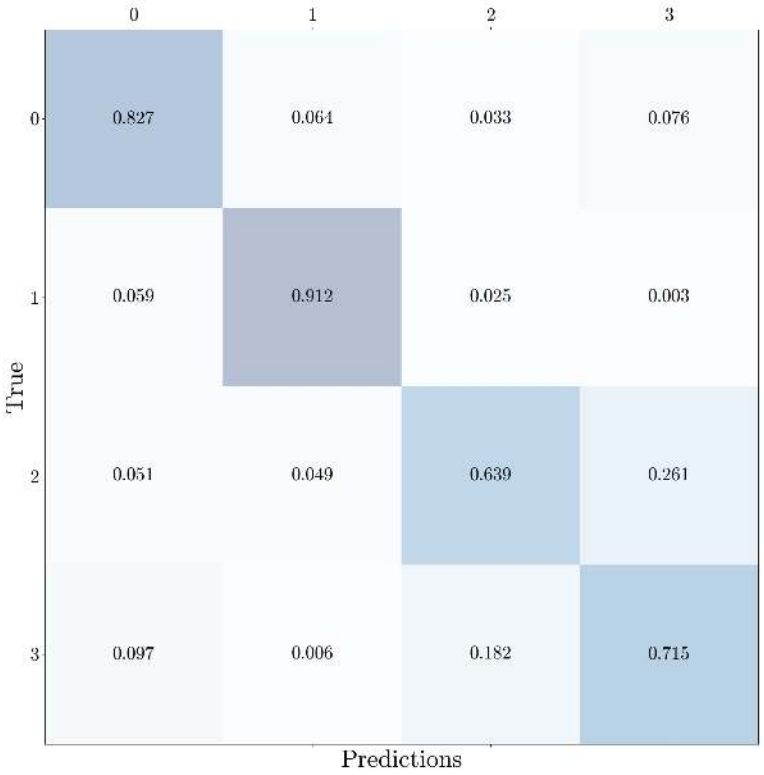


Figure 6.8: This confusion matrix shows the distribution of class predictions for our static late fusion results.

Interestingly, both fusion results fail to outperform the static MAEST embeddings. Since the late fusion results are close to the MAEST results, we chose to conduct an ablation study in which we remove different inputs. This study can be seen in Table 6.8.

As you can see, when removing the MAEST embeddings, the performance drops to a score similar to the static MERT embeddings. When the MERT embeddings are removed and only the handcrafted and Wav2Vec 2.0 embeddings remain, the resulting scores are the only ones that outperform their individual components. It

Feature Subset	**Macro F_1**	**Weighted F_1**
Hand., MERT, Wav2Vec 2.0, MAEST	77.16	77.34
Hand., MERT, Wav2Vec 2.0	74.08	74.33
Hand., Wav2Vec 2.0	70.62	70.90

Table 6.8: Ablation results from late fusion of different subsets of static feature inputs.

appears that the F_1 scores are constrained by the most informative embeddings which carry the most weight throughout the network, dominating the other inputs.

6.4 Regression

Our last experiment involves using continuous valence and arousal values for regression rather than discrete quadrant labels for classification. Similar to the DEAM dataset, the MERGE dataset provides static regression values. These values were obtained during the dataset's creation when converting the AllMusic tags into continuous valence and arousal values using Warriner's adjective list [50]. However, besides the baselines provided in the original MERGE paper [34], there are no studies that use these continuous representations. This includes the two later papers by Louro et al. [33] [35] in which they only use quadrant classification, not regression.

The regression baselines provided in the original MERGE paper for the Audio Complete dataset were obtained using their handcrafted feature set and an SVM. With this technique, they obtained R^2 correlation values of 0.479 and 0.364, and RMSE of 0.281 and 0.375 for arousal and valence respectively. For the Bimodal

Feature Type	**RMSE**		**r**		**R^2**	
	V	**A**	**V**	**A**	**V**	**A**
Static	**0.090**	**0.155**	**0.741**	**0.720**	**0.549**	0.519
Dynamic	0.092	0.158	0.733	0.711	0.533	0.506
Baseline [34]	0.305	0.169	–	–	0.505	**0.528**

Table 6.9: Results from our comparison of dynamic and static MAEST embedding on continuous regression targets. Our audio only static results represent the state-of-the-art performance on MERGE, outperforming the bimodal baseline.

Complete version they were able to increase these values to 0.505, 0.528 R^2 and 0.169, 0.305 RMSE. These results were obtained using Wav2Vec 2.0 audio embeddings combined with RoBERTa lyric embeddings. These two modalities were concatenated together and fed into a fully connected network with two hidden layers.

For our study, we utilize static and dynamic MAEST audio embeddings, and the same model architecture from our previous experiments (MLP for static, CNN-BLSTM + attention pooling for dynamic). The differences between our classification and regression models include the use of mean squared error instead of cross entropy loss, and the final fully connected layer which outputs two values (valence and arousal) and utilizes a sigmoid activation instead of softmax. This allows the model to output valence and arousal values between 0-1. The results of our regression study can be seen in Table 6.9. As shown, we outperform the baselines on all metrics.

Finally, when viewing these regression results, it is important to remember that they are the averaged results across twenty independent trials, or seeds. This

Metric	**Valence**	**Arousal**
RMSE	0.0009	0.0011
r	0.0056	0.0042

Table 6.10: The standard deviation of our static regression results from all 20 independent trails.

was done to ensure individual seeds were not producing outlier results, and for a direct comparison with the baselines. However, downstream applications would only utilize one seed, so it is also important to make sure there is not major fluctuations between our trials. To ensure this is not occurring, we calculate the standard deviation across all of our trials, and can be seen in Table 6.10. This table shows that our results are consistent and our state-of-the-art audio only model is not only reproducible, but would be useful in real world applications.

Chapter 7: Discussion

The experiments we conducted comprised an extensive comparison of dynamic versus static features and embeddings on music emotion classification. In this section, we analyze our results and provide insights on potential limitations and future work.

7.1 Dynamic vs. Static

Our initial hypothesis was that the dynamic sequential inputs would outperform static representations. This was due to the fact that the sequential features offer more information on how the songs change over time, and that temporal variation may contain emotionally relevant information. Additionally, with sequential features, stronger models and techniques like BLSTMs and attention pooling could be utilized to handle this information. However, this is not what we observed in our experiments. Instead, contrary to our intuition, the static features and models consistently outperformed the dynamic ones with the best-performing dynamic results being a weighted F_1-Score of 76.72% and the best static results of 77.78%.

While this could be the result of optimization error, as there are many more hyperparameters with stronger CNN-BLSTM models, we believe this is due to a mismatch in resolution between the sequential features and the static labels. The

global emotion labels are a high-level summary of the song's affective content. Dynamic features may contain too much noise and too many short-term fluctuations that are not relevant to the global label. As such, static aggregated features help emphasize the most relevant cues to determine the correct class. This could also help to explain why the MAEST embeddings performed so well as MAEST is a supervised model trained with a global-level label. This does not mean that there is no emotional information to be gained from the time axis or that the emotion in a song cannot change. It means that the static label does not provide enough information to support learning temporal variation.

These results, when viewed in the context of other music emotion classification studies, suggest an interesting conclusion. Previous work performed by the creators of the MERGE dataset and active contributors to MER research Panda et al. [41], [42] and Louro et al. [35], [33], [34] have consistently used static features and delegated music emotion variation detection to the side. Despite this, they continue to produce competitive results on MERGE and the older 4Q dataset [41]. Additionally, our state-of-the-art results on the 4Q dataset were obtained with an MLP model utilizing 300 static features [29].

However, despite this, other studies report comparable, if not better, results when utilizing dynamic features on other datasets. Many early studies utilized dynamic features with encoders similar to ours [14] [30], and transformer models have also been shown to be effective [20]. Additionally, many of the dynamic features are the result of Mel Spectrogram inputs which by their image nature, necessitate dynamic-level features. It is highly unlikely that the songs in MERGE and

4QAED do not work well with dynamic features compared to the songs in datasets like DEAM, MTG Jamendo, and MagnaTagATune. Yet, the differences between these datasets could be key to identifying why. DEAM contains dynamic-level labels as well as static global labels, and MTG Jamendo, and MagnaTagATune utilize multi-label tagging. Dynamic features benefit from these label types because the additional label information can better correspond to temporal features. Multilabel classification can also benefit from dynamic features as the change in emotion of a song can be represented by multiple tags instead of one static label.

As a result, we believe there is a need for a modernized MER dataset with dynamic labels similar to DEAM. With this, stronger emotional models can benefit from the subtle change in emotion. While DEAM does have dynamic labels, there are many well-known issues with the dataset including: high class imbalance, poor quality and unverified annotations, and questionable annotation acquisition. Many of these issues were the reasons why MERGE and 4Q datasets were created in the first place.

Our experiments also support the view that sequential features contain important emotional information. When comparing our sequential CNN-BLSTM model with a time-distributed fully connected model which does not model recurrent dependencies, we saw an increase in performance as seen in Table 6.4. Additionally, in our static experiments when comparing the average of all tokens with the classification tokens alone from MAEST, the average of all tokens (including sequential, classification, and distillation tokens) outperformed the static classification tokens, as seen in Table 6.5. This means the sequential tokens contain additional emotional

information not captured by the classification tokens alone. This is reinforced by our dynamic experiments with the MAEST sequence tokens in which we produced comparable results with only the sequential tokens, as seen in Table 6.2.

A final benefit of dynamic labels would be greater robustness to overfitting than is possible with static features. In previous experiments done with MERGE and 4Q [34] [33] [35] [41], ReliefF feature selection was performed to obtain the most important features before they are input to their models. This feature selection, when performed on a large number of statistically aggregated features (mean, median, std, min, max, skew, kurtosis) can lead to overfitting, especially evident with smaller sized datasets. This can be seen in our own experiments with our static handcrafted feature set. In our previous study on the 4QAED dataset (which only has 900 samples) we utilized a similar feature selection procedure to obtain the top three hundred features from a large set of aggregated features. In this study we obtained a macro F_1-Score of 82.03% with an MLP. However, when using the same static feature set here on the MERGE dataset with 3455 samples with the same MLP model we only obtain 68.39% macro F_1 [29]. This suggests that these 300 features were substantially overfit to the small 4QAED dataset. When utilizing sequential features, this type of overfitting would be alleviated as all the statistical features would be captured in the sequence.

7.2 Fusion

Although none of the fusion experiments resulted in an increase in performance, for either static or dynamic features, several interesting results emerge. When examining the late static results, the ablation study revealed that the F_1-Score appeared to be bounded by the strongest input type. For example, when all features were present, late fusion failed to outperform the MAEST only results. When the MAEST features were removed, the results were comparable to the MERT only results. Only when combining the handcrafted and Wav2Vec 2.0 features was a slight increase in performance seen. These results suggest that the most informative features carry the most weight and dominate the others. This can be understood as evidence that our different input types are not truly different modalities, but multiple views of the same underlying signal. While this experiment does not directly prove this, it implies that the pre-trained models and extracted features contextualize similar information. However, this information is still relevant. In a study by Kong et al., they demonstrated the ability of music transformer models trained under contrastive self-supervised learning (similar to Wav2Vec 2.0 and MERT) to learn both global and local musical properties, including representations like key estimation and beat tracking with no specific constraints to do so [24].

When examining the cross fusion results, it is evident that they failed to outperform other fusion methods as well as individual features. This could reinforce the idea stated above that the features encode similar information, or the different feature types are not unique enough to supply additional context to one another.

Finally, when looking at the early fusion results, the dimensionality of the input vector becomes so large that the curse of dimensionality may be affecting performance. Since there are more than 3,000 dimensions, the space becomes so large and the data so sparse, that it is difficult to create accurate boundaries.

7.3 Regression

Finally, our results on the static regression values are consistent with our call for a music emotion dataset with dynamic labels. Like our classification results, our regression results outperform the best audio-only results so far, and even outperform their audio and lyrics model. With this in mind, we argue that it is not necessary to discretize the regression values into the four quadrants of the Valence and Arousal graph. Although more fine-grained regression labels may be harder to predict than four classes, we have shown that it is possible. Additionally, as seen in Figure 6.5 which shows the predicted and true regression targets, we can see that valence is modeled better than arousal, and while we obtain the state-of-the-art regression metrics, it is visually not perfect. Furthermore, an affine transformation could be applied to the valence dimension to obtain even better results. Overall, more precise predictions will not only make MER applications more useful, but it will also help to advance the field as a whole. Finally, regression values would be much easier to implement for dynamically labeled datasets.

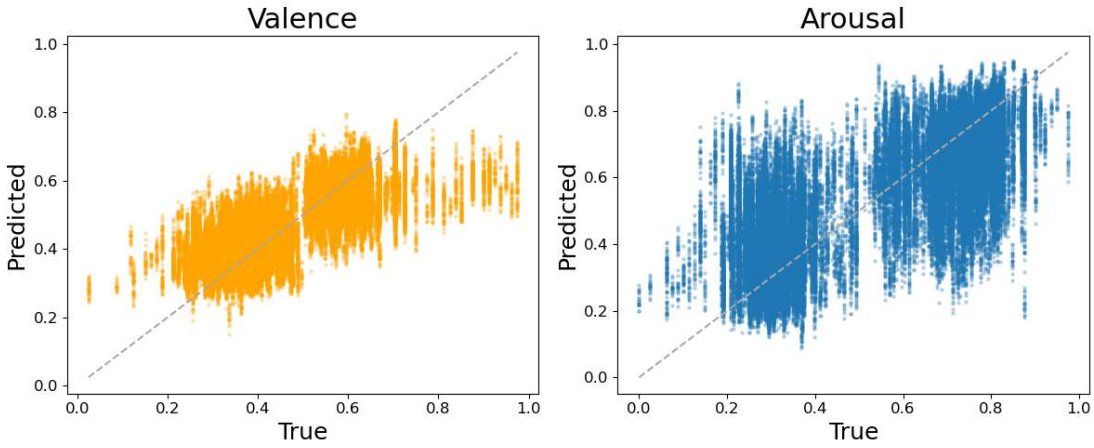


Figure 7.1: The distribution of our regression predictions when utilizing static MAEST embeddings.

7.4 Limitations

Despite our contributions, some limitations remain. First, the MERGE dataset does not contain dynamically annotated songs. As a result, we are unable to provide a complete comparison to the original authors of dynamic versus static features on both static and dynamically labeled data. This makes it difficult to properly evaluate the potential advantages of dynamic feature representations. While our results show that dynamic features do not improve results for statically labeled data, it is possible that a model trained using dynamic labels could still outperform our results here. While dynamically labeled datasets do exist (DEAM [1]), it is known to have many drawbacks including poor data annotation from annotator disagreement, large class imbalance, and many obscure royalty-free songs which may not generalize well to other music.

We find the results from the transformer model shown in Table 6.4 particularly interesting. It would seem a good idea to employ transformer architectures on a task like ours, which involves global labeling. The transformer layers can capture global emotional signals through self-attention. However, our experiments utilizing transformer blocks underperformed our CNN-BLSTM models. This may be consistent with the idea that adding more global-context capacity does not guarantee improvement. In our experiments, our blocks were simply added on top of the existing pre-trained transformer model MAEST with no fine-tuning. With our dataset of less than 4000 samples, adding more capacity can increase variance and may not be necessary. This limits the expressiveness of downstream models without fine-tuning.

Finally, we chose not to fine-tune because the goal of this study was not to maximize task performance for each pre-trained backbone, but to compare the static and dynamic representations under a controlled setting. Allowing end-to-end training would introduce an additional experimental factor beyond the central question of this work. Since Wav2Vec 2.0, MERT, and MAEST differ substantially in architecture, parameter count, and training strategy, fine-tuning could benefit each model unevenly and make the comparison less controlled. By keeping the pre-trained models frozen, we isolate the representations and reduce the influence of model specific optimization thereby maintaining a more reproducible experimental setup.

7.5 Future Work

In future work, we can address the limitations of this study, as well as continue to expand MER research. The main extension would be to utilize a dynamically annotated dataset like DEAM, and compare the results of models trained on dynamic features and changing labels. Since DEAM may be unreliable, an important step would be to either quality control the dataset to reduce some of the drawbacks, or create an entirely new dataset in a manner similar to MERGE but with dynamic labels.

Additionally, in order to gain increased performance on the MERGE dataset, the MAEST model could be fine-tuned for emotion classification. Although this was not done in this study as obtaining the highest F_1-score was not our main objective, fine-tuning MAEST specifically for MER could enhance its ability to capture affective content.

As a last step, incorporating lyric embeddings from models like ModernBERT or RoBERTa would help achieve the highest performance on MERGE. Louro et al. have already shown that the fusion of these two modalities greatly increases performance, especially in distinguishing valence when arousal is low [35]. The type of fusion of these two modalities could be an interesting experiment in a manner similar to the fusion experiments in this study.

Chapter 8: Conclusion

Through our experiments comparing dynamic and static music representations, we find, contrary to our initial hypothesis, that static features outperform dynamic features. In our work, we achieve the best audio-only performance on the MERGE dataset using static MAEST embeddings, with a weighted F_1 score of 77.78%. We also demonstrate that regression is a viable alternative to the simpler four-quadrant classification task. Notably, our audio-only results in both classification and regression are competitive with bimodal audio-and-lyric approaches reported in [35].

Our finding that static features outperform dynamic ones indicates that, for tasks involving a global-level emotion label, temporally aggregated information may be sufficient to capture the predominant emotional character of a song. As shown by our experiments, the additional temporal resolution provided by dynamic features does not translate into improved classification performance, and may introduce unnecessary noise. Despite this, we still achieve the best audio-only performance on the MERGE Complete dataset with a weighted F_1-Score of 77.78%. This was obtained with static embeddings from the MAEST model, and evaluated using an MLP with two hidden layers. Additionally, we test the regression version of the dataset and achieve the state-of-the-art performance there as well with an RMSE of 0.090 and 0.155 for valence and arousal, respectively. These

audio only results outperform the bimodal audio-and-lyric embeddings reported by Louro et al. [34]. This highlights MER's capacity to move away from simple four-class classification problems, and toward more challenging regression tasks.

With these results, we argue that an improved version of the MERGE dataset could be developed. This dataset would contain time-varying regression values that describe how the emotion of a song changes over time. Such a dataset would be highly beneficial for advancing MER, as it would allow models to learn the temporal evolution of emotion rather than a static label. A temporally annotated dataset would also reveal how emotional perception changes across different sections of a song. Sections such as verses, choruses, and bridges can evoke different emotions, and this labeling approach would allow these differences to be represented and potentially improve interpretability. Dynamic models and features could then be evaluated more appropriately and potentially reveal advantages not seen under static labeling. This would also encourage the development of models that capture long-term context and emotional transitions, an area that remains underexplored in current MER research.

Beyond the world of research, such a dataset would allow for many practical improvements. For example, many songs have drastic changes in tone and emotion that often define them. Music recommendation systems based on time-varying emotion prediction could better account for songs that transition from one emotion to another, enabling more fine-grained suggestions. Similarly, in music generation, a user would be able to describe how a song transitions from one emotion to another, or explicitly specify emotions at certain times. This would give them more

control over how their song is generated. Finally, in affective music therapy, the progression of emotion within the audio would allow for more gradual transitions from one emotion to another.

Bibliography

[1] Anna Aljanaki, Yi-Hsuan Yang, and Mohammad Soleymani. Developing a benchmark for emotional analysis of music. *Plos One*, 2017.

[2] Pablo Alonso-Jiménez, Xavier Serra, and Dmitry Bogdanov. Efficient supervised training of audio transformers for music representation learning, 2023.

[3] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[5] Laura-Lee Balkwill and William Thompson. A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues. *Music Perception*, 17:43–64, 10 1999.

[6] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José Zapata, and Xavier Serra. Essentia: an open-source library for sound and music analysis. In *Proceedings of the 21st ACM International Conference on Multimedia (ACM-MM)*, MM '13, page 855–858, New York, NY, USA, 2013. Association for Computing Machinery.

[7] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States, 2019.

[8] Densil Cabrera, Sam Ferguson, Farhan Rizwi, and Emery Schubert. Psysound3: a program for the analysis of sound recordings. *The Journal of the Acoustical Society of America (JASA)*, 123:3247, 06 2008.

[9] Yangyuan Chen, Zhizhong Ma, Mingjing Wang, and Mingzhe Liu. Advancing music emotion recognition: A transformer encoder-based approach. In

Proceedings of the 6th ACM International Conference on Multimedia in Asia, MMAsia '24, New York, NY, USA, 2024. Association for Computing Machinery.

[10] Yu-An Chen, Yi-Hsuan Yang, Ju-Chiang Wang, and Homer Chen. The AMG1608 dataset for music emotion recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 693–697. IEEE, 2015.

[11] Shreyan Chowdhury, Andreu Vall, Verena Haunschmid, and Gerhard Widmer. Towards explainable music emotion recognition: The route via mid-level features. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[12] Adyasha Dash and Kathleen Agres. Ai-based affective music generation systems: A review of methods and challenges. *ACM Comput. Surv.*, 56(11):1–34, July 2024.

[13] Rémi Delbouys, Romain Hennequin, Francesco Piccoli, Jimena Royo-Letelier, and Manuel Moussallam. Music mood detection based on audio and lyrics with deep neural net. In *International Conference on Music Information Retrieval (ISMIR)*, 09 2018.

[14] Yizhuo Dong, Xinyu Yang, Xi Zhao, and Juan Li. Bidirectional convolutional recurrent sparse network (bcrsn): An efficient model for music emotion recognition. *IEEE Transactions on Multimedia*, 21(12):3150–3163, 2019.

[15] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, pages 169–200, 1992.

[16] Alicia Fernández-Sotos, Antonio Fernández-Caballero, and José Miguel Latorre. Influence of tempo and rhythmic unit in musical emotion regulation. *Frontiers in Computational Neuroscience*, 10, 2016.

[17] Aalf Gabrielsson and Erik Lindström. The role of structure in the musical expression of emotions. *Handbook of music and emotion: Theory, research, applications*, pages 367–400, 01 2010.

[18] Shaurya Gaur and Patrick Donnelly. Mood dynamic playlist: Interpolating a musical path between emotions using a knn algorithm. In *Iternational Conference on Human Iteraction & Emerging Technologies: Artificial Intelligence*, 2022.

[19] Ziyi Huang, Minjie Wu, Xiao Shao, Mingwei Xu, and Chenyu Wang. Cross-level feature fusion for music emotion recognition: Integrating mel-spectrogram, high-level musical descriptors and symbolic midi data with attention mechanisms. In *2025 4th International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology (AIoTC)*, pages 124–134, 2025.

[20] Hanchen Jiang and Qin Zhang. Dual-stream transformer with psychoacoustic synergy for dynamic music emotion recognition. In *Submitted to International Conference on Machine Intelligence Theory and Applications*, 2025. under review.

[21] Xinxin Jiao, Liejun Wang, and Yinfeng Yu. Mfhca: Enhancing speech emotion recognition via multi-spatial fusion and hierarchical cooperative attention, 2024.

[22] Patrik N. Juslin. Perceived emotional expression in synthesized performances of a short melody: Capturing the listener's judgment policy. *Musicae Scientiae*, 1(2):225–256, 1997.

[23] Min-Jae Kim, Jeong-Geun Kim, Su-Kyung Yoon, and Shin-Dug Kim. Functionality-based processing-in-memory accelerator for deep convolutional neural networks. *IEEE Access*, pages 1–1, 10 2021.

[24] Yuexuan Kong, Gabriel Meseguer-Brocal, Vincent Lostanlen, Mathieu Lagrange, and Romain Hennequin. Emergent musical properties of a transformer under contrastive self-supervised learning. In *26th International Society for Music Information Retrieval (ISMIR)*, 2025.

[25] Olivier Lartillot, Petri Toiviainen, and Tuomas Eerola. A matlab toolbox for music information retrieval. In Christine Preisach, Hans Burkhardt, Lars Schmidt-Thieme, and Reinhold Decker, editors, *Data Analysis, Machine Learning and Applications*, pages 261–268, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[26] Edith Law, Kris West, Michael Mandel, Mert Bay, and J. Downie. Evaluation of algorithms using games: The case of music tagging. In *10th International Society for Music Information Retrieval (ISMIR)*, pages 387–392, 01 2009.

[27] Yoonhyung Lee, Seunghyun Yoon, and Kyomin Jung. Multimodal speech emotion recognition using cross attention with aligned audio and text. In *Interspeech 2020*, Interspeech 2020, page 2717–2721. ISCA, October 2020.

[28] Jianjun Lei, Kai Ye, and Ying Wang. Cross-attention fusion for audio-visual emotion recognition with shared transformer. *Speech Communication*, 178:103356, 2026.

[29] Shaun Leib. Interpreting emotion from sound: A comparison of features and models for music emotion classification. Master's thesis, Oregon State University, 2025.

[30] Xinxing Li, Jiashen Tian, Mingxing Xu, Yishuang Ning, and Lianhong Cai. DBLSTM-based multi-scale fusion for dynamic emotion prediction in music. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2016.

[31] Xinxing Li, Haishu Xianyu, Jiashen Tian, Wenxiao Chen, Fanhang Meng, Mingxing Xu, and Lianhong Cai. A deep bidirectional long short-term memory based multi-scale approach for music dynamic emotion prediction. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 544–548. IEEE, 2016.

[32] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu. Mert: Acoustic music understanding model with large-scale self-supervised training, 2024.

[33] Pedro Lima Louro, Hugo Redinho, Ricardo Malheiro, Renato Panda, and Rui Pedro Paiva. Improving music emotion recognition by leveraging hand-crafted and learned features. In *1st Latin American Music Information Retrieval Workshop*, 2024.

[34] Pedro Lima Louro, Hugo Redinho, Ricardo Santos, Ricardo Malheiro, Renato Panda, and Rui Pedro Paiva. Merge – a bimodal audio-lyrics dataset for static music emotion recognition, 2025.

[35] Pedro Lima Louro, Tiago Ribeiro, Ricardo Malheiro, Renato Panda, and Rui Pedro Paiva. Bee-mer: Bimodal embeddings ensamble for music emotion recognition. *Sound and Music Computing Conference (SMC)*, 7 2025.

[36] Lie Lu, D. Liu, and Hong-Jiang Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio, Speech, and Language Processing (TASLPRO)*, 14(1):5–18, 2006.

[37] Ye Ma, XinXing Li, Mingxing Xu, Jia Jia, and Lianhong Cai. Multi-scale context based attention for dynamic music emotion prediction. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1443–1450, New York, NY, USA, 2017. Association for Computing Machinery.

[38] Miroslav Malík, Sharath Adavanne, Konstantinos Drossos, Tuomas Virtanen, Dasa Ticha, and Roman Jarina. Stacked convolutional and recurrent neural networks for music emotion recognition. In *Sound and Music Computing Conference (SMC)*, volume 14, 07 2017.

[39] Benoit Mathieu, Slim Essid, Thomas Fillon, Jacques Prado, and Gael Richard. Yaafe, an easy to use and effcient audio feature extraction software. In *11th International Society for Music Information Retrieval Confrence (ISMIR)*, 2010.

[40] Charles Osgood, George Soci, and Percy TannenBaum. The dimensionality of the semantic space. *The Measurement of Meaning*, pages 1–30, 1957.

[41] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 11(4):614–626, 2020.

[42] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. Audio features for music emotion recognition: A survey. *IEEE Transactions on Affective Computing*, 14(1):68–88, 2023.

[43] Robert Plutchik. A general psychoevolutionary theory of emotion. *Theories of Emotion*, pages 3–33, 1980.

[44] Jordi Pons, Thomas Lidy, and Xavier Serra. Experimenting with musically motivated convolutional neural networks. In *14th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2016.

[45] James Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178, 12 1980.

[46] Ricardo Scholz, Geber Ramalho, and Giordano Cabral. Cross task study on mirex recent results: an index for evolution measurement and some stagnation hypotheses. In *17th International Society for Music Information Retrieval Conference (ISMIR)*, 08 2016.

[47] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[48] Xiaohan Shi, Xingfeng Li, and Tomoki Toda. Multimodal fusion of music theory-inspired and self-supervised representations for improved emotion recognition. *Interspeech*, 2024.

[49] George Tzanetakis and Perry Cook. Marsyas: a framework for audio analysis. *Organised Sound*, 4:169–175, 09 2002.

[50] Amy Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45:1191–1207, 02 2013.

[51] Bin Wu, Andrew Horner, and Chung Lee. The correspondence of music emotion and timbre in sustained musical instrument sounds. *Journal of the Audio Engineering Society (AES)*, 62:663–675, 10 2014.

[52] Tien-Lin Wu and Shyh-Kang Jeng. Probabilistic estimation of a novel music emotion model. In Shin'ichi Satoh, Frank Nack, and Minoru Etoh, editors, *Advances in Multimedia Modeling*, pages 487–497, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[53] Dan Yang and Won-Sook Lee. Disambiguating music emotion using software agents. In *5th International Conference on Music Information Retrieval (ISMIR)*, 2004.

[54] Yi-Hsuan Yang, Yu-Ching Lin, Ya-Fan Su, and Homer H. Chen. A regression approach to music emotion recognition. *IEEE Transactions on Audio, Speech, and Language Processing (TASLPRO)*, 16(2):448–457, 2008.

[55] Shaode Yu, Jiajian Meng, Wenqing Fan, Ye Chen, Bing Zhu, Hang Yu, Yaoqin Xie, and Qiurui Sun. Speech emotion recognition using dual-stream representation and cross-attention fusion. *Electronics*, 13(11), 2024.

[56] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270, 2019.

[57] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision (ECCV)*, pages 818–833, Cham, 2014. Springer International Publishing.

[58] Ruoyu Zhao, Xiantao Jiang, F. Richard Yu, Victor C. M. Leung, Tao Wang, and Shaohu Zhang. Leveraging cross-attention transformer and multifeature fusion for cross-linguistic speech emotion recognition. *IEEE Internet of Things Journal*, 12(23):50653–50664, 2025.